

Herausgeber

F. HOFFMANN

E. HÜLLERMEIER

R. MIKUT



Dortmund | 24. – 25. November 2016

PROCEEDINGS **26. WORKSHOP**
COMPUTATIONAL INTELLIGENCE



Scientific
Publishing

F. Hoffmann, E. Hüllermeier, R. Mikut (Hrsg.)

Proceedings. 26. Workshop Computational Intelligence

Dortmund, 24. – 25. November 2016

PROCEEDINGS **26. WORKSHOP** COMPUTATIONAL INTELLIGENCE

Dortmund, 24. – 25. November 2016

Herausgegeben von

F. Hoffmann

E. Hüllermeier

R. Mikut

Impressum



Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark of Karlsruhe
Institute of Technology. Reprint using the book cover is not allowed.

www.ksp.kit.edu



*This document – excluding the cover, pictures and graphs – is licensed
under the Creative Commons Attribution-Share Alike 3.0 DE License
(CC BY-SA 3.0 DE): <http://creativecommons.org/licenses/by-sa/3.0/de/>*



*The cover page is licensed under the Creative Commons
Attribution-No Derivatives 3.0 DE License (CC BY-ND 3.0 DE):
<http://creativecommons.org/licenses/by-nd/3.0/de/>*

Print on Demand 2016

ISBN 978-3-7315-0588-4
DOI 10.5445/KSP/1000060007

Inhaltsverzeichnis

S. Lu, E. Hüllermeier (Paderborn University) Support Vector Classification on Noisy Data using Fuzzy Superset Losses	1
J. Á. González Ordiano, W. Doneit, S. Waczowicz, L. Gröll, R. Mikut, V.Hagenmeyer (Karlsruhe Institute of Technology, Institute for Applied Computer Science) Nearest-Neighbor Based Non-Parametric Probabilistic Forecasting with Applications in Photovoltaic Systems	9
M. Alsayegh, C. Rösmann, F. Hoffmann, T. Bertram (TU Dortmund) Model Predictive Control for Learning from Demonstration	31
J. Schneider (Universität Osnabrück) Robuste globale Approximatoren durch Aufspaltung der Transformationsfunktion	43
J. H. Schoenke (Universität Osnabrück) Skalierbare inkrementelle On-line Regression auf Simplex-Strukturen	51
M. Krüger, S. Meuresch, A. Stockem Novo, T. Nattermann, K.H. Glander, T. Bertram (ZF TRW, TU Dortmund) Strukturanalyse eines neuronalen Netzes zur Spurwechselprädiktion für das automatisierte Fahren	57

S.Chandrasekaran, M.Zaefferer, S.Moritz, J.Stork, M.Friese, A.Fischbach, T.Bartz-Beielstein	77
(TH Köln)	
Data Preprocessing: A New Algorithm for Univariate Imputation Designed Specifically for Industrial Needs	
A. Kroll	97
(Universität Kassel)	
Zum optimalen Testsignalentwurf für die Parametrierung von Partitionierung und Teilmodellen dynamischer Takagi-Sugeno- Modelle: Problemstellung und Lösungsansätze	
M. Gringard, A. Kroll	119
(Universität Kassel)	
On the systematic parametrization of APRBS and multisine test signals for nonlinear system identification	
T. O. Heinz, O. Nelles	139
(Universität Siegen)	
Vergleich von Anregungssignalen für Nichtlineare Identifikationsaufgaben	
A. Fischbach, M. Zaefferer, J. Stork, M.Friese, T. Bartz-Beielstein	159
(TH Köln)	
From Real World Data to Test Functions	
M. Trommer, A. Wenzel, C.Walther	179
(Hochschule Schmalkalden, Fraunhofer IOSB)	
Neuronale Netze vs. Support-Vektor-Maschinen – Ein direkter Vergleich der Klassifikationsmethoden für komplexe biologische Daten	
M.Schneider, C.Walther, A.Wenzel	193
(HS Schmalkalden, Fraunhofer IOSB AST)	
Classification of Production Quality in Injection Moulding with an Embedded Diagnostic System Using a Fuzzy Inference System	
C. Dengler	205
(TU München)	
Entwurf und Vergleich nichtlinearer Beobachter am realen Viertanksystem, basierend auf Takagi-Sugeno-Modellen	

T. Aissa, A. Cavaterra, S. Lambeck (Hochschule Fulda) Takagi-Sugeno Fuzzy Systeme für örtlich verteilte Raumklimagrößen	225
B. Jäschke, A. Kroll (Universität Kassel, FB Maschinenbau, FG Mess- und Regelungstechnik) Ein Nächste-Nachbarn-Ansatz zur Anomaliedetektion bei Massendaten aus kontinuierlich betriebenen Chemieanlagen	245
T. Loose (Hochschule Heilbronn) Reglerentwurf mit klassischen und Computational Intelligence Methoden in der Hochschullehre am Beispiel von Modellen	261

Support Vector Classification on Noisy Data using Fuzzy Superset Losses

Shenzhen Lu and Eyke Hüllermeier¹

¹Department of Computer Science
Paderborn University, Germany
E-Mail: lushenzhen@gmail.com, eyke@upb.de

Abstract

We apply a recent method for so-called *superset learning* to the problem of binary classification with support vector machines on noisy data. More specifically, we propose a generalization of the hinge loss based on the concept of (fuzzy) optimistic superset loss. This generalization allows for modulating the influence of individual training examples and can be seen as an alternative to standard instance weighting; in particular, it increases the robustness toward noise and outliers in the data. Experimentally, our method compares favorably with other approaches for learning from noisy data.

1 Superset Learning

Superset learning is a specific type of learning from weak supervision, in which the outcome (response) associated with a training instance is only characterized in terms of a subset of possible candidates. Thus, superset learning is somehow in-between supervised and semi-supervised learning, with the latter being a special case (in which supersets are singletons for the labeled examples and cover the entire output space for the unlabeled ones). There are numerous applications in which only partial information about outcomes is available [7]. Correspondingly, the superset learning problem has received increasing attention and has been studied by various authors in recent years, albeit under different names [2, 6, 9, 1].

Consider a standard setting of supervised learning with an input (instance) space \mathcal{X} and an output space \mathcal{Y} . The goal is to learn a mapping from \mathcal{X}

to \mathcal{Y} that captures, in one way or the other, the dependence of outputs (responses) on inputs (predictors). The learning problem essentially consists of choosing an optimal model (hypothesis) M^* from a given model space (hypothesis space) \mathbf{M} , based on a set of training data

$$\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N \in (\mathcal{X} \times \mathcal{Y})^N . \quad (1)$$

More specifically, optimality typically refers to optimal prediction accuracy, i.e., a model is sought whose expected prediction loss or *risk*

$$\mathcal{R}(M) = \int L(y, M(\mathbf{x})) d\mathbf{P}(\mathbf{x}, y) \quad (2)$$

is minimal; here, $L : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ is a loss function, and \mathbf{P} is an (unknown) probability measure on $\mathcal{X} \times \mathcal{Y}$ modeling the underlying data generating process.

Here, we are interested in the case where output values $y_n \in \mathcal{Y}$ are not necessarily observed precisely; instead, only a superset Y_n of y_n is observed, i.e., a set $Y_n \subseteq \mathcal{Y}$ such that $y_n \in Y_n$. Therefore, the learning algorithm does not have direct access to the (precise) data (1), but only to the (imprecise, ambiguous) observations

$$\mathcal{O} = \{(\mathbf{x}_n, Y_n)\}_{n=1}^N \in (\mathcal{X} \times 2^{\mathcal{Y}})^N . \quad (3)$$

2 Generalized Loss Minimization

Recall the principle of *empirical risk minimization* (ERM): A model M^* is sought that minimizes the *empirical risk*

$$\mathcal{R}_{emp}(M) = \frac{1}{N} \sum_{n=1}^N L(y_n, M(\mathbf{x}_n)) , \quad (4)$$

i.e., the average loss on the training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. The empirical risk (4) serves as a surrogate of the true risk (2). In order to avoid the problem of possibly *overfitting* the data, not (4) itself is typically minimized but a *regularized* version thereof.

In [5], we introduced an approach to superset learning that is motivated by the idea of performing model identification and “data disambiguation”

simultaneously. This idea is realized by means of a generalized risk minimization approach, using an extended loss function that compares precise predictions with set-valued observations. If $L : \mathcal{Y}^2 \rightarrow \mathbb{R}_+$ is the original loss, the *optimistic superset loss* is defined as follows:

$$L^* : 2^{\mathcal{Y}} \times \mathcal{Y} \rightarrow \mathbb{R}_+, (Y, \hat{y}) \mapsto \min \{L(y, \hat{y}) \mid y \in Y\} \quad (5)$$

In [4], we propose a further generalization to the case where data is characterized by fuzzy subsets $\tilde{Y} \in \mathcal{F}(\mathcal{Y})$:

$$L^{**} : \mathcal{F}(\mathcal{Y}) \times \mathcal{Y} \rightarrow \mathbb{R}_+, (\tilde{Y}, \hat{y}) \mapsto \int_0^1 L^*([\tilde{Y}]_\alpha, \hat{y}) d\alpha, \quad (6)$$

where $[\tilde{Y}]_\alpha$ is the α -cut of \tilde{Y} .

In addition to learning from genuinely imprecise or ambiguous data, our framework for superset learning can also be used for learning from standard (precise) data, which—via a process of “imprecisiation”—is deliberately turned into imprecise data. In this way, different effects can be achieved. In particular, data imprecisiation offers a means to modulate the influence of individual observations on the overall result of the learning process: the more imprecise an observation is made, the less it will influence the model induced from the data.

In [8], we applied our framework to *locally weighted* linear and support vector regression, showing that a method that emerges as a natural instance of our framework is competitive to standard approaches. In this paper, we turn our attention to (binary) *classification* or, more specifically, support vector classification [10]. Actually, this seems to be an even more interesting case, mainly because the generalized loss function (6) suggested by our framework (obtained by replacing an observed class label by a fuzzy subset of classes) deviates more drastically from the (weighted) hinge loss that is commonly used in support vector classification—see Figure 1 for an illustration.

3 Support Vector Classification with Fuzzy Loss Functions

Consider the case of binary classification and let $\mathcal{Y} = \{-1, +1\}$ encode the two classes (negative and positive). Moreover, suppose that \mathbf{M} is a class of *scoring classifiers* $M : \mathcal{X} \rightarrow \mathbb{R}$; a positive score $s = M(\mathbf{x}) > 0$ suggests

that \mathbf{x} belongs to the positive class, whereas a negative score suggests that \mathbf{x} is negative. A margin loss is a function of the form

$$L(y, s) = f(y s) , \tag{7}$$

where $f : \mathbb{R} \rightarrow \mathbb{R}$ is non-increasing. Thus, a margin loss penalizes scores instead of binary predictions, and the larger (smaller) the score in the case of a positive (negative) class, the smaller the loss. An important example of (7) is the hinge loss

$$L(y, s) = f(y s) = \max(1 - y s, 0) \tag{8}$$

used in support vector machines [10].

Now, suppose that the output is characterized by a fuzzy subset \tilde{Y} of \mathcal{Y} , that is, by a membership degrees $\mu_{\tilde{Y}}(-1)$ and $\mu_{\tilde{Y}}(+1)$ for the negative and positive class, respectively. More specifically, suppose

$$\mu_{\tilde{Y}}(\lambda) = \begin{cases} 1 & \text{if } \lambda = y \\ 1 - w & \text{if } \lambda = \bar{y} \end{cases} , \tag{9}$$

where $\{y, \bar{y}\} = \{-1, +1\}$ and w can be interpreted as a degree of confidence in y . Then, it is not difficult to show that the fuzzy loss function (6) is given by

$$\mathbb{L}(Y, s) = f_w(y s) = w \cdot f(y s) + (1 - w) \cdot f(|y s|) . \tag{10}$$

Please note that f_w coincides with the original margin loss f if $y s > 0$, i.e., if the prediction $s = M(\mathbf{x})$ is in favor of the more likely class y ; thus, the difference only concerns the negative part. Figure 1 (left) shows the graph of (10) for different values of w .

Especially interesting is the observation that, in the case of full uncertainty (maximal imprecision), our loss coincides with the so-called *hat loss*, which is commonly used in semi-supervised support vector classification; interpreting the absence of any label information as complete ignorance, this makes perfect sense. Note that, despite not expressing any preference in favor of one of the classes, the hat loss still encourages a margin effect. In contrast to this, the weighted hinge loss simply vanishes (and yields 0 everywhere), which means that unlabeled data points have no influence on the training process anymore.

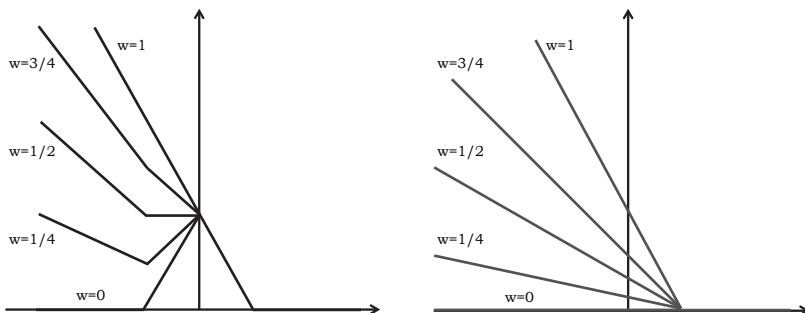


Figure 1: Left: Generalized (fuzzy) loss function (6) for different weighings w of the training example. Right: Standard version of weighted hinge loss, obtained by multiplying the original loss with w .

4 Experiments

We empirically compare our method with other robust loss minimization techniques [3] in the setting of noisy data, i.e., classification problems in which the observed class labels are corrupted with noise. In addition to standard SVM, the following methods are included in the experiments: Robust truncated-hinge-loss support vector machines (RSVM) trains SVMs with the a truncated version of the hinge loss in order to be more robust toward outliers and noisy data [11]. The one-step weighted SVM (OWSVM) first trains a standard SVM. Then, it weighs each training example based on its distance to the decision boundary and retrains using the weighted hinge loss shown on the right in Figure 1 [12].

Our approach (FLSVM) is the same as OWSVM, except for the weighted loss: instead of using a simple weighting of the hinge loss, we use the optimistic fuzzy loss (Figure 1, left); the value w in (10) is obtained through an exponential weighting of the distance from the decision problem. This approach requires solving a non-convex optimization problem, for which we implemented the so-called concave-convex procedure [13].

The data sets Wisconsin Breast Cancer (569 instances, 31 attributes) and Bupa Liver Disorder (345 instances, 7 attributes) were used in [11, 12]. In addition, we used Banknote Authentication (1372 instances, 5 attributes) and Parkinson’s Disease Detection (195 instances, 23 attributes).

Table 1: Experimental results: Average misclassification rate on test data (with standard deviation) for different methods, data sets, and noise levels.

perc	data sets	SVM	OWSVM	RSVM	FLSVM
0%	Wdbc	0.0281 (0.0114)	0.0263 (0.0087)	0.0228 (0.0100)	0.0374 (0.0159)
	Bupa	0.3188 (0.0928)	0.3043 (0.0774)	0.3072 (0.0776)	0.3188 (0.0934)
	Banknote	0.0153 (0.0110)	0.0095 (0.0050)	0.0153 (0.0083)	0.0124 (0.0101)
	Parkinsons	0.1333 (0.0334)	0.1128 (0.0292)	0.1077 (0.0215)	0.1077 (0.0215)
10%	Wdbc	0.0387 (0.0133)	0.0281 (0.0144)	0.0316 (0.0146)	0.0334 (0.0130)
	Bupa	0.3391 (0.0442)	0.3304 (0.0527)	0.3159 (0.0635)	0.3159 (0.0720)
	Banknote	0.0233 (0.0063)	0.0168 (0.0110)	0.0146 (0.0068)	0.0131 (0.0095)
	Parkinsons	0.1385 (0.0229)	0.1231 (0.0215)	0.1231 (0.0215)	0.1179 (0.0215)
20%	Wdbc	0.0615 (0.0124)	0.0474 (0.0100)	0.0386 (0.0171)	0.0422 (0.0209)
	Bupa	0.3855 (0.0364)	0.3478 (0.0369)	0.3275 (0.0873)	0.3362 (0.0601)
	Banknote	0.0241 (0.0050)	0.0248 (0.0056)	0.0211 (0.0125)	0.0175 (0.0075)
	Parkinsons	0.1385 (0.0466)	0.1333 (0.0493)	0.1279 (0.0429)	0.1436 (0.0493)
30%	Wdbc	0.0791 (0.0270)	0.0633 (0.0245)	0.0633 (0.0314)	0.0580 (0.0302)
	Bupa	0.3884 (0.0854)	0.3710 (0.0861)	0.3826 (0.1033)	0.3710 (0.1018)
	Banknote	0.0313 (0.0110)	0.0277 (0.0077)	0.0270 (0.0092)	0.0255 (0.0215)
	Parkinsons	0.1846 (0.0459)	0.1897 (0.0693)	0.1846 (0.0712)	0.1692 (0.0716)

The training data is contaminated by randomly flipping the labels with probabilities 0, 0.1, 0.2, 0.3. We use 5×5 (internal) cross validation for parameter tuning and evaluation.

As can be seen, all robust methods tend to be better than the standard SVM. Our approach appears to be better than OWSVM, which makes use of standard weighting. Moreover, it is at least competitive to RSVM, with a trend toward better results with an increasing level of noise.

5 Conclusion

Our first experimental results are clearly encouraging and confirm that (fuzzy) superset learning is an interesting alternative to standard weighting in learning from noisy data. Needless to say, more intensive empirical studies are needed before profound conclusions can be drawn.

Another topic of future work is the complexity of our approach, which requires solving a non-convex optimization problem. A reduction of the runtime, which is still significantly higher than that of standard SVM, would be highly desirable.

References

- [1] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12:1501–1536, 2011.
- [2] Y. Grandvalet. Logistic regression for partial labels. In *IPMU-02, Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 1935–1941, Annecy, France, 2002.
- [3] B. Han, I.W. Tsang, and L. Chen. On the convergence of a family of robust losses for stochastic gradient descent. In *Proc. ECML/PKDD*, Riva del Garda, Italy, 2016.
- [4] E. Hüllermeier. Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization. *International Journal of Approximate Reasoning*, 55(7):1519–1534, 2014.
- [5] E. Hüllermeier and W. Cheng. Superset learning based on generalized loss minimization. In *Proceedings ECML/PKDD-2015, European Conference on Machine Learning and Knowledge Discovery in Databases*, Porto, Portugal, 2015.
- [6] R. Jin and Z. Ghahramani. Learning with multiple labels. In *16th Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, 2002.
- [7] L.P. Liu and T.G. Dietterich. A conditional multinomial mixture model for superset label learning. In *Proc. NIPS*, 2012.
- [8] S. Lu and E. Hüllermeier. Locally weighted regression through data imprecisiation. In *Proceedings 25. Workshop Computational Intelligence*, pages 97–104, Dortmund, Germany, 2015. KIT Scientific Publishing.
- [9] N. Nguyen and R. Caruana. Classification with partial labels. In *Proc. KDD 2008, 14th Int. Conf. on Knowledge Discovery and Data Mining*, Las Vegas, USA, 2008.
- [10] B. Schölkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.
- [11] Yichao Wu and Yufeng Liu. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102(479), 2007.

- [12] Yichao Wu and Yufeng Liu. Adaptively weighted large margin classifiers. *Journal Comput. Graph. Stat.*, 22(2), 2013.
- [13] A.L. Yuille and A. Rangarajan. The concave-convex procedure. In *Proc. NIPS*, 2002.

Nearest-Neighbor Based Non-Parametric Probabilistic Forecasting with Applications in Photovoltaic Systems

Jorge Ángel González Ordiano, Wolfgang Doneit, Simon Waczowicz, Lutz Gröll, Ralf Mikut, Veit Hagenmeyer

Karlsruhe Institute of Technology, Institute for Applied Computer Science
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen
E-Mail: jorge.ordiano@kit.edu

1 Introduction

Time series forecasting (i.e. the prediction of unknown future time series values using known data) has found several applications in a number of fields, like, economics and electricity forecasting [15]. Most of the used forecasting models deliver a so-called point forecast [4], a value that according to the models' criteria is most likely to occur. Nonetheless, such forecasts lack information regarding their uncertainty. A possibility of quantifying such uncertainty is by conducting probabilistic forecasts [6, 10], which can be delivered as prediction intervals (including the probability of the forecast being inside the interval) or complete conditional probability distributions of future time series values [12]. Such a quantification of the forecast uncertainty is of interest for several optimization problems, as e.g. model predictive control. Probabilistic forecasting is divided in parametric and non-parametric approaches. While the former assume that the forecast values follow a known distribution (e.g. Gaussian) and try to determine the parameters describing it, the latter make no assumptions, but instead attempt to approximate the underlying distribution via the training data. Non-parametric approaches have the advantage of not assuming that all values will follow the same probability distribution across all points in time [3, 21].

Quantile regression is an approach that has been utilized in several works (e.g. Bremnes [1] and Nielsen et al. [17]) in order to create non-parametric probabilistic forecasts. It is a type of regression that provides the conditional quantile of an output value given a certain input, and it is obtained through

the minimization of a non-differentiable loss-function [5]. By combining several of such quantile regression models (i.e. point forecasting models), probabilistic forecasts can be conducted [14]. However, the creation and implementation of quantile regressions, especially with more complex data-driven regression techniques, as artificial neural networks, is not a simple task, due to its non-differentiable loss function [2]. For this reason, the present contribution offers a simple method of obtaining quantile regressions, which can be combined in data-driven probabilistic forecasts, by changing the models' training set using a k-nearest-neighbor approach, traditional regression methods, and the assumption of a dense dataset.

The potential of the presented methodology for the obtainment of probabilistic forecasting models is demonstrated for the usecase of photovoltaic (PV) power forecasting. The periodicity and length of the utilized PV power time series helps to assure the dataset's density that is necessary for the present contribution's method. Accurate PV forecasting models are of major importance, due to the PV systems' dependence on the weather that makes their electrical power generation uncertain. Such uncertainty complicates the balancing of demand and supply in the electrical grid [20].

The present contribution is divided as follows: First, the principles of forecasting models are described. Then, the general framework of the data-driven non-parametric nearest-neighbor based probabilistic forecasting approach is presented. Afterwards, the principles behind the present work's photovoltaic power forecasting approach as well as the utilized dataset are shown. The next section shows and discusses the obtained results. Finally, a conclusion and an outlook are offered.

2 Forecasting Models

Data-driven point forecasting models are used to approximate the values of a desired time series y (whose discrete time steps are denoted by $k \in [1, K]$) at a forecast horizon H (with $H \geq 1$) using its current or past values and/or values of other exogenous time series, \mathbf{u} . For example, the functional relation given by a linear or non-linear autoregressive exogenous (ARX or NARX) forecasting model that utilizes the current and past information from timestep k to timestep $k - H_1$ of its input time series and whose

parameters are represented by a vector θ , can be written as¹

$$\hat{y}[k + H] = f(y[k], \dots, y[k - H_1], \mathbf{u}^T[k], \dots, \mathbf{u}^T[k - H_1]; \theta); k > H_1 . \quad (1)$$

The obtainment of data-driven point forecasting models can be generalized as a regression problem, i.e. the models are obtained through a learning process in which the value of a model's loss function in a training set is minimized. For example, models obtained by minimizing the sum of squared errors (as is the case with typical data mining techniques, like polynomial regression or artificial neural networks) deliver as forecast an approximation of the conditional expected value given their used input [8, 12].

A probabilistic forecast can be attained by combining several point forecasting models which instead of delivering the conditional expected value, deliver an approximation of a conditional q -quantile, $\hat{y}_q[k + H]$ (with $q \in [0.01; 0.99]$). The functional relation of such models, using the same input as in Equation (1), is written in the present contribution as

$$\hat{y}_q[k + H] = f_q(y[k], \dots, y[k - H_1], \mathbf{u}^T[k], \dots, \mathbf{u}^T[k - H_1]; \theta_q) . \quad (2)$$

Just as in Equation (1), the vector θ_q represents the parameters defining the model. Equation (2) describes a quantile regression and it is obtained through the minimization of the sum of the so-called pinball-loss [1, 5], which is given as

$$L_q = \sum_{k=H_1+1}^{K-H} \begin{cases} (q - 1) (y[k + H] - \hat{y}_q[k + H]) & \text{if } y[k + H] < \hat{y}_q[k + H] \\ q (y[k + H] - \hat{y}_q[k + H]) & \text{if } y[k + H] \geq \hat{y}_q[k + H] \end{cases} . \quad (3)$$

Unfortunately, the pinball-loss is non-differentiable, that makes its minimization with standard algorithms and data mining techniques a complicated task. For such reason, the present contribution offers an alternative that allows the creation of quantile regression models using traditional approaches.

3 Method

The present section describes the principles behind the developed k-nearest-neighbor based non-parametric probabilistic forecasting approach. Those

¹The functional relation in Equation (1) can also describe autoregressive (AR) models, if the exogenous time series \mathbf{u} are not used as input values.

principles are divided into how the required data-driven quantile regressions are obtained and how those are combined forming intervals which can be used as probabilistic forecasting models. Additionally, the evaluation values, utilized in the present contribution to determine the accuracy of the created models, are presented. As previously stated, the creation of point forecasting models (i.e. the quantile regressions) can be generalized into a regression problem, hence, the methodology description is formulated in a general manner.

3.1 k-Nearest-Neighbor Based Quantile Regression

In order to solve a regression problem a training set containing a series of learning pairs is required. The training set is comprised of N desired output values, y_n , and their corresponding input vectors $\mathbf{x}_n^T = (x_{n1}, \dots, x_{nS})$, with S being the number of its contained features. The learning pairs are normally written as an input matrix \mathbf{X} of dimensions $N \times S$ and a N dimensional desired output vector \mathbf{y} :

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix}, \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}. \quad (4)$$

Both \mathbf{x}_n^T and y_n can be defined for the time series case shown in Equation (1) as

$$\begin{aligned} \mathbf{x}_n^T &= (y[n + H_1], \dots, y[n], \mathbf{u}^T[n + H_1], \dots, \mathbf{u}^T[n]), \\ y_n &= y[n + H + H_1], \text{ for} \\ n &\in [1, \dots, N], \text{ with } N = K - H - H_1. \end{aligned} \quad (5)$$

The approach in the present contribution assumes that each input vector has several similar neighbors in the training set, i.e. a dense input space. Hence, the nearest neighbors' corresponding output values should more or less represent the conditional distribution of the possible outputs given the input vector in question (periodic time series, as the ones utilized in the present work, help to assure in some degree the correctness of such an assumption). The present contribution uses such property in order to change the learning pairs depending on the q -quantile regression ($q \in [0.01; 0.99]$) which is going to be created. The value q represents the nominal coverage of the desired quantile regression, meaning that if $q = 0.8$ and the regression is perfect then 80% of all y -values will always be less than the values of the obtained model.

The procedure starts by calculating the distances between the i^{th} learning pair's input and all the others,

$$d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j) \forall i, j \in [1; N], \quad (6)$$

with $d(\cdot, \cdot)$ as the distance function utilized for their calculation (which can change depending on the model's application). The distance function utilized in the present work is described in one of the following sections.

The obtained distances allow the determination of the i^{th} input's k-nearest-neighbors (k-NN), whose corresponding output values are sorted in ascending order. Thereafter, the values in the sorted k-NNs output set are set equal to the

$$\left(\frac{0.5}{k_{NN}}, \frac{1.5}{k_{NN}}, \dots, \frac{k_{NN} - 0.5}{k_{NN}} \right) \quad (7)$$

quantiles, with k_{NN} being the number of used nearest-neighbors. If the desired q -quantile is equal to one of those, then its value is taken from the sorted k-NNs output set, if not its value is interpolated from the nearest-neighbors' output set - the described procedure is given in the works of Hyndman [11] (defined as Definition 4) and Langford [13] (referred to as Method 10). Afterwards, the i^{th} output value is substituted by the obtained quantile. The process is repeated until a new set of desired outputs, \mathbf{y}_q , is created:

$$\mathbf{y}_q = \begin{pmatrix} y_{q,1} \\ \vdots \\ y_{q,N} \end{pmatrix}. \quad (8)$$

Data mining techniques can then be trained using the changed training set and conventional regression approaches, thus delivering a model that approximates in average the conditional quantile of the desired outputs given the used input values and defined by the amount of the utilized nearest-neighbors. The above described methodology has the advantage of obtaining a functional relation for the wanted conditional quantiles using only the nearest-neighbors during the training procedure and hence, eliminating the necessity of saving the training set and of conducting the nearest-neighbors calculation during the models' usage.

For the sake of illustration, Figure 1 shows the principles behind the present contribution's method for the case of a one dimensional input vector and a linear regression model.

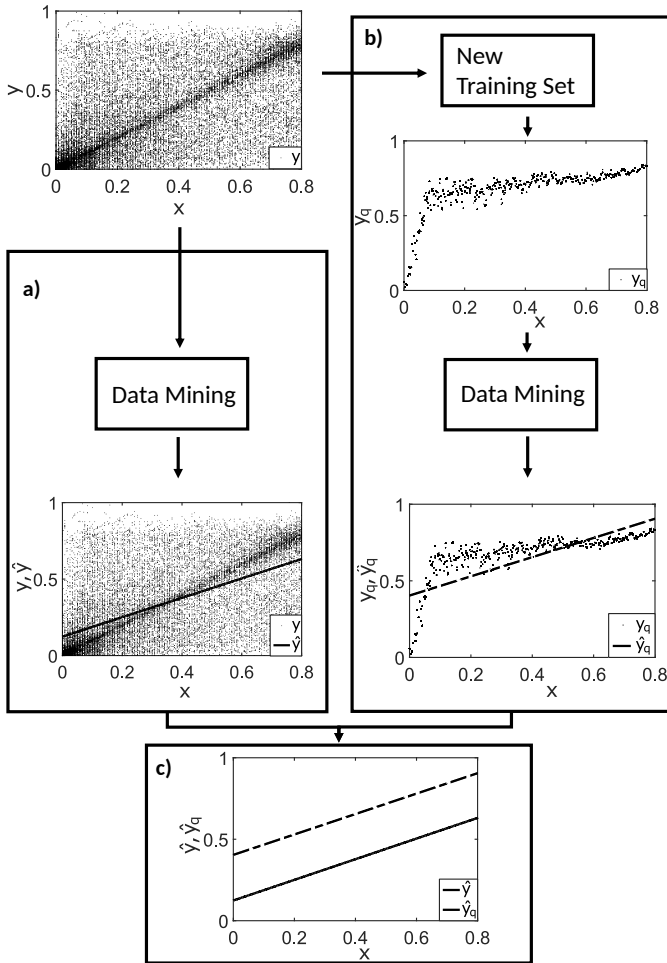
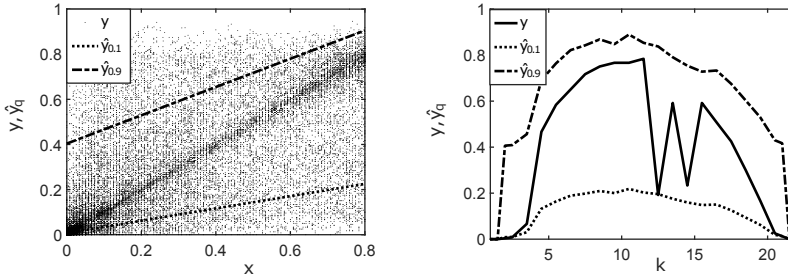


Figure 1: Nearest-neighbor based quantile regression principles using linear regression models

- a) Regression model obtained using the unchanged training set
- b) Regression model obtained using the changed training set for $q = 0.9$
- c) Comparison of the two different regression models



(a) Quantile regression in feature space (b) Quantile regression as time series

Figure 2: Example of interval forecast using pairs of linear quantile regressions

3.2 Quantile Regression Intervals

By combining pairs of quantile regressions, intervals with a desired nominal coverage can be created. For example, an interval with a nominal coverage of 0.8, i.e. an interval in which 80% of all y -values will lay inside (if the interval model is perfect), can be obtained by using the 0.9 and the 0.1-quantile regressions; with the former as the upper interval bound (\hat{y}_{q_u}) and the latter as the lower interval bound (\hat{y}_{q_l}). The combination of both quantile regressions is given as:

$$\hat{\mathbf{y}}_{(q_u - q_l)} = \begin{bmatrix} \hat{y}_{q_u} \\ \hat{y}_{q_l} \end{bmatrix}, \quad (9)$$

with $(q_u - q_l)$ representing the difference between the nominal coverage of the upper and lower bounds and thus, the nominal coverage of the created interval.

In the present contribution, the regressions' desired output values are future values of a time series, so the created intervals can be utilized as interval forecasting models. An illustrative example of linear quantile regressions forming the upper and lower bounds, when represented as regression models (for a one-dimensional input) and as time series, is shown in Figure 2.

3.3 Evaluation

Traditional evaluation values like mean absolute error or root mean square error are not adequate to determine the accuracy of quantile regressions and the respectively created intervals. Therefore, other kinds of evaluation values have been proposed in literature.

The first value used in the present work is the reliability [9, 12, 18, 21]. It describes the actual coverage of a quantile regression or an interval, i.e. the percentage of values which actually lay under the former and inside the latter. Due to the fact, that a common way of depicting the accuracy of quantile regressions or intervals is by plotting their reliability deviation (i.e. the difference between their actual coverage and their nominal coverage) against their nominal coverage, the reliability deviation ΔRl is chosen as the present contribution's evaluation value. Such value is given for a single quantile regression and a testset with N samples as

$$\Delta Rl_q = \frac{1}{N} \text{card}_{n=1, \dots, N} (y_n < \hat{y}_{q,n}) - q, \quad (10)$$

while for an interval formed by quantile regressions as

$$\Delta Rl_{(q_u - q_l)} = \frac{1}{N} \text{card}_{n=1, \dots, N} (\hat{y}_{q_l, n} \leq y_i < \hat{y}_{q_u, n}) - (q_u - q_l); \quad (11)$$

with $\text{card}(\cdot)$ being the cardinality operator. The more the reliability deviation approximates zero, the better the evaluated model is. A negative value describes an underestimating quantile regression or a narrow interval, while a positive value points at the contrary. Furthermore, the reliability deviation does not quantify the extent in which the measurements violate the quantile regression or the interval bounds, but instead treats every violation equally. A problem that arises by using the reliability deviation is, that if a quantile regression or an interval underestimate in some cases, but overestimate in others by the same amount, then their reliability deviation will result in a perfect regression or interval. However, such result is incorrect.

Due to the above mentioned problems other evaluation values presented by Gneiting et. al [6] are also used in the present work. First, in order to evaluate the quality of quantile regressions the pinball-loss is used. The pinball-loss for a single quantile regression and a testset with N samples is given as

$$L_q = \text{mean}_{n=1, \dots, N} \{(y_n - \hat{y}_{q,n})(q - I(y_n < \hat{y}_{q,n}))\}, \quad (12)$$

with $I(\cdot)$ being the indicator function². As it can be seen in Equation (12) the pinball-loss considers the magnitude of each deviation and weights it depending on the quantile regression being evaluated. The lower its value is, the better the evaluated regression is. A similar value, referred in the present work as interval's pinball-loss, is used for the evaluation of the created intervals. Such value can only be utilized if the condition $q_u = 1 - q_l$ holds and it is given as

$$L_{(q_u - q_l)} = \text{mean}_{n=1, \dots, N} \{ (\hat{y}_{q_u, n} - \hat{y}_{q_l, n}) + \frac{2}{1 - (q_u - q_l)} (y_n - \hat{y}_{q_u, n}) I(y_n > \hat{y}_{q_u, n}) + \frac{2}{1 - (q_u - q_l)} (\hat{y}_{q_l, n} - y_n) I(y_n < \hat{y}_{q_l, n}) \} . \quad (13)$$

The interval's pinball-loss has as its first term the distance between its bounds, thus considering a narrow interval better than a broader one and allowing the identification of unwanted trivial intervals. A trivial interval is, for example, an interval with a nominal coverage of 0.99 for y -values ranging between 0 and 1 formed by the two constant values 0 and 0.99. The second and third term quantify the amount in which the values outside of the interval deviate from it; those deviations are weighted depending on the interval's nominal coverage. The lower the pinball-loss value is, the better the evaluated interval is.

4 Photovoltaic Power Forecasting

The goal of the present contribution's probabilistic photovoltaic power forecasting is to obtain accurate quantile regressions with $q \in [0.01; 0.99]$ for a forecast horizon of $24h$ utilizing information of the past generated power, as well as the previously described nearest-neighbor based approach. Those regressions are obtained with various amounts of nearest-neighbors (50, 70, 100, and 120) for the training set transformations. After their creation they are combined into probabilistic forecasting models. The present section describes the data and techniques utilized for creating the presented PV power forecasting models.

²Function which equals 1 if its condition is fulfilled and 0 otherwise.

4.1 Dataset

The dataset utilized in the present contribution is a freely available dataset provided by the Australian energy provider Ausgrid³. After the dataset's preprocessing (which is conducted just as in [7]), time series containing measurements of the normalized average generated photovoltaic power (normalized to values between 0 and 1) every 15 minutes for the time frame of July 1st, 2010 to June 30th, 2013 ($K = 52608$) of 54 rooftop PV systems are obtained. The 54 rooftop PV systems comprise the clean dataset defined by Ratnam et al. [19]. The dataset does not contain historical nor forecast weather information, hence all forecasting models described in the following sections are created using purely the information of the past generated power. Furthermore, the utilized time series are separated in halves, with the first half used as training set and the second as test set.

4.2 Input Time Series

As already mentioned, the models shown in the present contribution only utilize information of the past generated power (P). From P , two other time series, which are also utilized as input, are created. The first one is given by the equation

$$P_{max}[k] = \max\{P[k], P[k - H_p], \dots, P[k - m \cdot H_p]\} , \quad (14)$$

with H_p being the number of time steps representing 24 hours (i.e. the day's periodicity) and $m = 7$. This new created time series contains the measured maximal values of the last eight days. Additionally, the second created time series, given by the equation

$$P_{mean}[k] = \text{mean}\{P[k], P[k - H_p], \dots, P[k - m \cdot H_p]\} , \quad (15)$$

contains the average values of the last eight days. The creation of the previously described time series has the goal of eliminating the time series' random effects, while retaining systematic repeating ones (e.g. shadowing effects caused by neighboring buildings), with P_{mean} also retaining information of the previous week's weather variability. The usage of maximal and mean time series to reduce the influence of past random effects is reserved for periodic time series, due to their repeating nature.

³<http://www.ausgrid.com.au>

With this time series, Equation (2) is extended to

$$\hat{P}_q[k + H] = f_q(P[k], \dots, P[k - H_1], P_{max}[k], \dots, P_{max}[k - H_1], P_{mean}[k], \dots, P_{mean}[k - H_1]; \theta_q). \quad (16)$$

In the present work, the values H , H_p , and H_1 are defined as 96 time steps (number of timesteps representing 24h due to the utilized time series' resolution).

4.3 Elimination of Night Values

In order to increase the accuracy of the probabilistic PV power forecasting models, night values are eliminated from the used training set. For the elimination, the following assumption is made: If the generated PV powers 24 and 48 hours prior to the forecast horizon ($H = 96$) are less or equal to a threshold, then both values as well as the value at the forecast horizon are most likely night values; thus both the value at the forecast horizon (desired output) and its corresponding input vector are eliminated from the training set. This assumption allows the creation of the forecasting models on the basis of non-trivial values. In the present contribution the threshold was set equal to 0.0001. During the quantile regressions' usage the same assumption is utilized. Therefore, only future values which are considered to be day values are forecast by the created models, all other values are set automatically equal to zero (night values). Likewise, the quantile regressions and the interval forecasting models are only evaluated on test set values considered to be day measurements.

4.4 Data Mining Techniques

The present contribution utilizes three different polynomial models without bilinear terms, two artificial neural networks, and a support vector regression to determine quantile regressions for $q \in [0.01; 0.99]$ with which the PV power interval forecasts are to be obtained. The polynomial models are referred to as Poly1, Poly2, and Poly3 and describe polynomials with a maximal degree of one, two, and three. The artificial neural networks (ANN) are multilayer perceptrons with a single hidden layer, but a different number of neurons in it; the first referred to as ANN6 has six neurons, while the second, ANN10, has ten neurons. Finally, the support vector regression (SVR) utilizes a Gaussian kernel of degree one, a trade off value

C equal to 1 and an ϵ equal to 0.01. All utilized models are created using the MATLAB toolbox Gait-CAD [16].

Before the calculation of the PV power quantile regressions is undertaken, the distance function used for the determination of the nearest-neighbors has to be defined. In the present work, the weighted Euclidean distance is utilized. The necessary weights are defined as the inverse of the features' variance in the used training set. Both the distance function and its weights are described as

$$d_{ij} = \left(\sum_{s=1}^S w_s (x_{is} - x_{js})^2 \right)^{1/2} \quad \forall i, j \in [1; N] , \quad (17)$$

$$w_s = (\text{var}(\mathbf{X}(:, s)))^{-1} \text{ for } s \in [1, S] .$$

For features which are constant, a regularization has to be conducted during its weight calculation in order to avoid dividing by zero.

Furthermore, from all input values shown in Equation (16), four of them are selected individually for each model type and each household to be utilized as features during the creation and application of the models, hence $S = 4$. The selection process assures that the only difference between the quantile regressions from a specific technique are not the utilized features, but the parameters θ_q obtained from their different training sets. Additionally, the selection of four features prior to the nearest-neighbors calculation reduces the possibility of sparsity in the used feature space, which in high dimensions would be inevitable - even when utilizing the past values of the periodic PV power time series as input. The selection procedure is a forward feature selection. Due to the long computational time required for the SVR models' creation, the forward feature selection is not applied for this type of model; instead, the features selected for ANN6 are used. According to the notation in Equations (4) and (8) the input vectors and their corresponding desired outputs, necessary for the application of the present work's method, are given as $\mathbf{x}_n^T = (x_{n1}, x_{n2}, x_{n3}, x_{n4})$ and $y_{q,n} = P_q[k + H]$.

The actual obtainment of the PV power polynomial quantile regressions starts by determining the polynomial describing the smallest quantile ($q = 0.01$) by minimizing the sum of squared errors and using the constraint that its values should be greater than zero. Afterwards, the remaining quantile regressions are created under the constraint that their values should be greater than the ones obtained by the previously obtained quantile regression model. This constraint avoids the problem of quantile crossing [5],

in which a quantile regression delivers values which are smaller than the ones provided by a model representing a lower quantile. Additionally the vector $\mathbf{1}_N$, which only contains ones, is added to the input matrix, hence allowing offsets in the desired polynomials, thus making them affine functions. Nonetheless, the distances in Equation (17) are still calculated using the original input matrix \mathbf{X} . The determination of the polynomials is given by the following equations:

$$\begin{aligned}
& \underset{\boldsymbol{\theta}_q}{\text{minimize}} && \|\mathbf{y}_q - [\mathbf{1}_N \ \mathbf{X}]\boldsymbol{\theta}_q\|^2 \\
& \text{subject to} && \begin{cases} [\mathbf{1}_N \ \mathbf{X}]\boldsymbol{\theta}_q \geq 0 & \text{if } q = 0.01 \\ [\mathbf{1}_N \ \mathbf{X}]\boldsymbol{\theta}_q \geq [\mathbf{1}_N \ \mathbf{X}]\boldsymbol{\theta}_{(q-0.01)} & \text{else} \end{cases} \quad (18) \\
& \text{with} && \boldsymbol{\theta}_q^T = (\theta_{q,0}, \theta_{q,1}, \theta_{q,2}, \theta_{q,3}, \theta_{q,4}) .
\end{aligned}$$

The greater than zero constraint introduced for the smaller quantile stems from the a-priori known fact that generated PV power should only be positive. The possibility of implementing all of the constraints with ease is the main reason why polynomial models are utilized in the present contribution. Of course, their implementation comes at the cost of making the assumption that the output conditional quantiles must change linearly in the given feature space.

The creation of the ANN6 and ANN10 models is conducted utilizing the Levenberg–Marquardt algorithm with a maximum of 20 training epochs, while the SVR models are obtained with the Gait-CAD implementation of the libsvm C++ library. The ANNs and SVR do not require the linearity assumption of the polynomials. However, they have the drawback of not using any constraints during the creation of each quantile regression. Therefore, in order to avoid quantile crossings, the following equation, with $\hat{y}_{q,n} = \hat{P}_q[k + H]$, has to be used during their evaluation:

$$\hat{y}_{q,n} = \begin{cases} \max(\hat{y}_{q,n}, 0) & \text{if } q = 0.01 \\ \max(\hat{y}_{q,n}, \hat{y}_{q-0.01,n}) & \text{else} \end{cases} . \quad (19)$$

The value $\hat{y}_{q,n} = \hat{P}_q[k + H]$ represents the corrected value after the application of the above equation.

Table 1: Average quantile regressions' reliability deviation on test set for $H = 96$

k_{NN}	Poly1 %	Poly2 %	Poly3 %	ANN6 %	ANN10 %	SVR %
50	5.93	5.66	5.89	4.44	4.24	1.60
70	5.95	5.68	5.95	4.28	3.87	1.59
100	5.95	5.67	5.93	4.07	3.61	1.63
120	5.94	5.66	5.91	3.81	3.47	1.68

Table 2: Average quantile regressions' pinball-loss on test set for $H = 96$

k_{NN}	Poly1 %	Poly2 %	Poly3 %	ANN6 %	ANN10 %	SVR %
50	4.21	4.21	4.22	4.14	4.14	4.11
70	4.21	4.21	4.22	4.14	4.14	4.11
100	4.21	4.21	4.22	4.14	4.14	4.12
120	4.21	4.21	4.22	4.13	4.14	4.12

5 Results

First of all, the quality of the present work's assumption to identify night values has to be addressed. It shows an accuracy of 99% when tested on the complete utilized dataset.

In order to evaluate the different quantile regressions, averages of the reliability deviation's absolute values and the pinball-loss on the utilized test set are presented in Tables 1 and 2. The given values are averages across the 54 used households and 99 quantile regressions created with each data mining technique and different amounts of nearest-neighbors. Furthermore, those values are obtained by evaluating the models on the test set values considered to be day measurements.

The contents in Table 1 show a clear improvement in the reliability deviation when utilizing the non-linear techniques, with SVR - especially SVR with $k_{NN} = 70$ - showing the lowest average reliability deviations. The average pinball-loss values in Table 2 also show the improvements of utilizing non-linear approaches. Nonetheless, the differences in the pinball-loss averages are not as remarkable as the ones shown in the reliability deviation results.

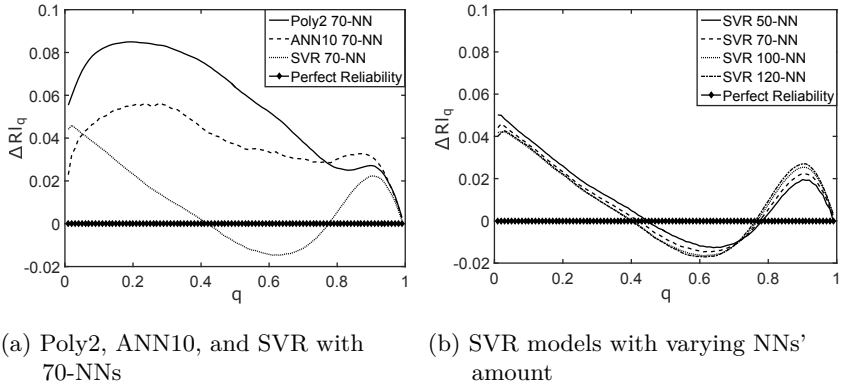


Figure 3: Test set reliability deviations of quantile regressions

The SVR with 70-NNs creates in average the best quantile regressions, for such reason, Figure 3a compares the reliability deviation for each nominal coverage to the ones obtained by the best polynomial (Poly2) and the best artificial neural network (ANN10) for $k_{NN} = 70$. Likewise Figure 3b plots the reliability deviation against the nominal coverage for the SVR models obtained using different amounts of NNs.

Figure 3a shows that SVR with $k_{NN} = 70$ obtains the models whose absolute reliability deviation is the lowest; with the only exception being the lowest nominal coverages in which ANN10 delivers regressions with a lower deviation. Likewise, it is possible to discern that both Poly2 and ANN10 overestimate in average, while SVR fluctuates between overestimating and underestimating depending on the considered nominal coverage. Interestingly, the non-linear ANN10 shows an advantage over Poly2 until a nominal coverage of approximately 0.8, afterwards, Poly2 becomes better. Figure 3b depicts how the increasing number of nearest-neighbors affects the accuracy. The figure shows that the lower the amount of nearest-neighbors is, the greater the reliability deviation for low quantiles (for q lower than 0.4) is. However, for higher quantiles (for q higher than 0.8) the opposite is true. The effect could be explained by the fact that using a higher amount of nearest-neighbors allow the lower quantiles to reflect effects like complete changes in weather (e.g. like rainy days after a couple of sunny days), but at the same time introduce output values corresponding to not so similar input vectors that skew the higher quantile regressions.

Table 3: Average intervals' reliability deviation on test set for $H = 96$

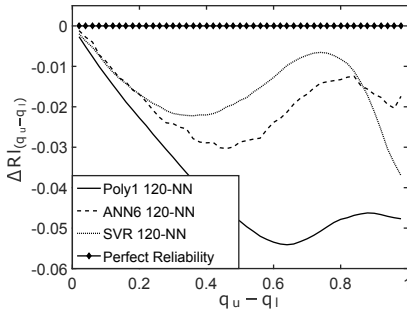
k_{NN}	Poly1 %	Poly2 %	Poly3 %	ANN6 %	ANN10 %	SVR %
50	4.57	4.51	4.76	2.48	2.15	2.15
70	4.24	4.21	4.47	2.31	1.86	1.84
100	3.98	3.99	4.26	1.80	1.69	1.61
120	3.86	3.90	4.19	1.91	1.58	1.56

Table 4: Average intervals' pinball loss on test set for $H = 96$

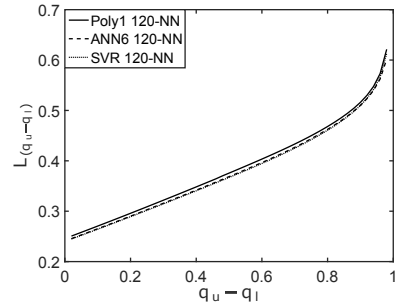
k_{NN}	Poly1 %	Poly2 %	Poly3 %	ANN6 %	ANN10 %	SVR %
50	38.56	38.54	38.66	37.93	37.95	37.72
70	38.54	38.53	38.65	37.90	37.96	37.73
100	38.53	38.52	38.64	37.91	37.90	37.77
120	38.53	38.51	38.64	37.88	37.92	37.80

By combining pairs of the obtained quantile regressions (starting with the regressions for $q_u = 0.51$ and $q_l = 0.49$ and ending with $q_u = 0.99$ and $q_l = 0.01$) 49 different interval forecasting models, with nominal coverage from 0.02 to 0.98, are created. Table 3 contains the reliability deviation average across all nominal coverage for the techniques' interval forecasting models. Likewise, Table 4 contains its average interval's pinball-loss values.

Just as before, the values contained in Tables 3 and 4 show an improvement through the usage of non-linear approaches. With SVR again as the technique delivering the most accurate models. The one using 120-NNs has the lowest reliability deviation, while the one utilizing 50-NNs possesses the overall best interval's pinball-loss. Figures 4a and 4b show both evaluation values across all nominal coverages for the Poly1, ANN6, and SVR with $k_{NN} = 120$, due to the fact that with such k_{NN} the majority of techniques obtain their most accurate models. Poly1 and ANN6 are chosen additionally, because they possess the lowest reliability deviation for their type. Furthermore, Figures 5a and 5b plot the evaluation values against nominal coverage for SVR with varying amounts of nearest-neighbors.

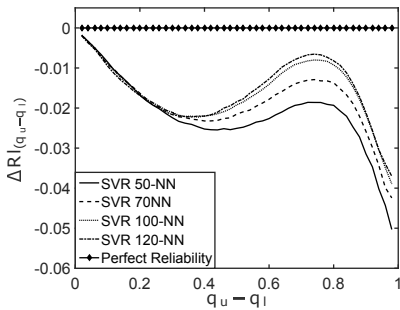


(a) Reliability deviation

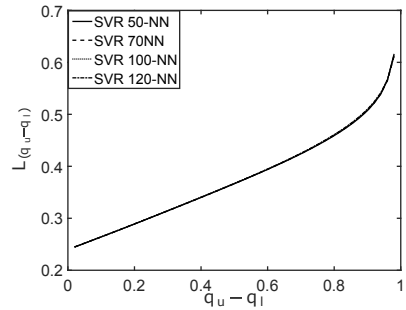


(b) Interval's pinball loss

Figure 4: Test set evaluation values of interval forecasts with 120-NNs



(a) Reliability deviation



(b) Interval's pinball loss

Figure 5: Test set evaluation values of SVR interval forecasts

Figures 4a and 4b exemplify how the non-linear approaches are able to deliver more accurate interval forecasting models independently of the desired nominal coverage. Additionally, Figure 4a shows that techniques create underestimating intervals independently of the desired nominal coverage; a property which can be attributed to the higher reliability deviation of the lower quantiles in comparison to the higher ones. Furthermore, Figure 5a depicts how the increasing number of nearest-neighbors appear to decrease the reliability deviations for a nominal coverage greater than approximately 0.3, while Figure 5b shows an independence of the interval's pinball-loss (at least for the SVR case) towards the amounts of nearest-neighbors used in the present contribution. Moreover, the fact that the interval's pinball-loss

for a nominal coverage higher than approximately 0.4 was lower than the nominal coverage shows that trivial interval forecast are not created. The reason behind the high pinball-loss values for intervals with a nominal coverage lower than 0.4 can be attributed to the great amount of values outside of them; by visualizing their results it can be concluded that non-trivial intervals are created. An example of obtained interval forecasts can be seen in Figure 6. It displays interval forecasts obtained with SVR and 120-NNs for a nominal coverage of 0.2, 0.4, 0.6, and 0.8.

Even though the SVR technique seems to provide the most accurate quantile regressions and interval forecasting models, its long training times pose a major drawback. For example, the creation of the 99 SVR quantile regressions without the forward feature selection using the present work’s method takes approximately *24min*, when utilizing a Intel Core i7-4790 3.60GHz processor and 16Gb of RAM. In comparison, the training of the same 99 quantile regressions with the forward feature selection takes approximately *1min* for the polynomial techniques, while for the artificial neural networks it takes approximately *12min*.

Finally as a last remark, it is important to mention that by using the present contribution’s methodology, the obtainment of quantile regressions with non-linear and complex data mining techniques (i.e. ANN and SVR) is simplified and made possible without changing the traditional algorithms used for their model’s training.

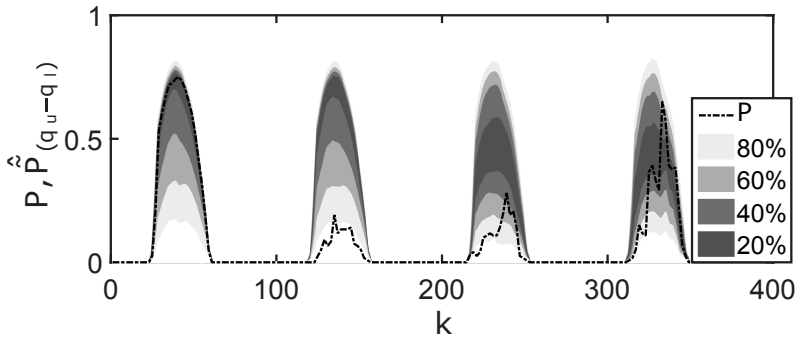


Figure 6: SVR 120-NNs interval forecasts for different nominal coverage (0.2, 0.4, 0.6, and 0.8)

6 Conclusion and Outlook

The present contribution offers a simple methodology for the obtainment of data-driven interval forecasting models by combining pairs of quantile regressions. Those regressions are created without the usage of the non-differentiable pinball-loss function, but through a k-nearest-neighbors based training set transformation and traditional regression approaches. By leaving the underlying training algorithms of the data mining techniques unchanged, the presented approach simplifies the creation of quantile regressions with more complex techniques (e.g. artificial neural networks). The quality of the presented methodology is tested on the usecase of photovoltaic power forecasting, for which quantile regressions using polynomial models as well as artificial neural networks and support vector regressions are created. From the resulting evaluation values it can be concluded that acceptable interval forecasting models are created.

It is important to mention, that all quantile regressions for a specific household, data mining technique, and number of nearest-neighbors, are created with the same selected features. For such reason, future works should examine how much does the quantile regressions' and interval forecasts' quality increases or decreases if each quantile regression is able to select its optimal features separately. Another aspect which requires further inquiry is that the interval forecasting models obtained from using the present contribution's method are only able to quantify the uncertainty of the model's output given some uncertainty-free input values. Therefore, research regarding the quantification and propagation of input values uncertainties (e.g. uncertainty in forecast weather data) and ways to differentiate the uncertainty coming from the input values and the one coming from the model itself has to be conducted further. Also, the testing of the presented methodology on a benchmark dataset has to be carried out in order to identify its shortcomings and develop further improvements. Additionally, future PV power related works should test the methodology with different forecast horizons as well as different inputs (e.g. forecast solar irradiation) for the purpose of testing its behavior and its quality.

Acknowledgments

The present contribution is supported by the Helmholtz Association under

Acknowledgments

The present contribution is supported by the Helmholtz Association under the Joint Initiative “Energy System 2050 - A Contribution of the Research Field Energy”.

References

- [1] BREMNES, J. B.: Probabilistic Wind Power Forecasts Using Local Quantile Regression. *Wind Energy* 7(1) (2004), pp. 47–54.
- [2] CANNON, A. J.: Quantile Regression Neural Networks: Implementation in R and Application to Precipitation Downscaling. *Computers & Geosciences* 37(9) (2011), pp. 1277–1284.
- [3] CHARYTONIUK, W.; CHEN, M.-S.; KOTAS, P.; VAN OLINDA, P.: Demand Forecasting in Power Distribution Systems using Nonparametric Probability Density Estimation. *IEEE Transactions on Power Systems* 14(4) (1999), pp. 1200–1206.
- [4] CHATFIELD, C.: Calculating Interval Forecasts. *Journal of Business & Economic Statistics* 11(2) (1993), pp. 121–135.
- [5] FAHRMEIR, L.; KNEIB, T.; LANG, S.: *Regression*. Statistik und ihre Anwendungen, Springer, 2009.
- [6] GNEITING, T.; KATZFUSS, M.: Probabilistic Forecasting. *Annual Review of Statistics and Its Application* 1 (2014), pp. 125–151.
- [7] GONZÁLEZ ORDIANO, J. Á.; WACZOWICZ, S.; REISCHL, M.; MIKUT, R.; HAGENMEYER, V.: Photovoltaic Power Forecasting using Simple Data-driven Models without Weather Data. *Computer Science - Research and Development* (2016), pp. 1–10.
- [8] HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2008.
- [9] HONG, T.; FAN, S.: Probabilistic Electric Load Forecasting: A Tutorial Review. *International Journal of Forecasting* (2016).
- [10] HONG, T.; PINSON, P.; FAN, S.; ZAREIPOUR, H.; TROCCHI, A.; HYNDMAN, R. J.: Probabilistic Energy Forecasting: Global Energy

Forecasting Competition 2014 and beyond. *International Journal of Forecasting* 32 (2016) 3, pp. 896 – 913.

- [11] HYNDMAN, R. J.; FAN, Y.: Sample Quantiles in Statistical Packages. *The American Statistician* 50(4) (1996), pp. 361–365.
- [12] JUBAN, J.; SIEBERT, N.; KARINIOTAKIS, G. N.: Probabilistic Short-Term Wind Power Forecasting for the Optimal Management of Wind Generation. In: *Proc., IEEE Lausanne Power Tech*, pp. 683–688, IEEE, 2007.
- [13] LANGFORD, E.: Quartiles in Elementary Statistics. *Journal of Statistics Education* 14(3) (2006), pp. 1–6.
- [14] MA, X.; HE, X.; SHI, X.: A Variant of K Nearest Neighbor Quantile Regression. *Journal of Applied Statistics* (2015), pp. 1–12.
- [15] MAIMON, O.; ROKACH, L.: *Data Mining and Knowledge Discovery Handbook*, Bd. 2. Springer, 2005.
- [16] MIKUT, R.; BURMEISTER, O.; BRAUN, S.; REISCHL, M.: The Open Source Matlab Toolbox Gait-CAD and its Application to Bioelectric Signal Processing. In: *Proc., DGBMT-Workshop Biosignalverarbeitung, Potsdam*, pp. 109–111, 2008.
- [17] NIELSEN, H. A.; MADSEN, H.; NIELSEN, T. S.: Using Quantile Regression to Extend an Existing Wind Power Forecasting System with Probabilistic Forecasts. *Wind Energy* 9(1-2) (2006), pp. 95–108.
- [18] PINSON, P.; NIELSEN, H. A.; MØLLER, J. K.; MADSEN, H.; KARINIOTAKIS, G. N.: Non-Parametric Probabilistic Forecasts of Wind Power: Required Properties and Evaluation. *Wind Energy* 10(6) (2007), pp. 497–516.
- [19] RATNAM, E. L.; WELLER, S. R.; KELLETT, C. M.; MURRAY, A. T.: Residential Load and Rooftop PV Generation: an Australian Distribution Network Dataset. *International Journal of Sustainable Energy* (2015), pp. 1–20.
- [20] WACZOWICZ, S.; REISCHL, M.; KLAIBER, S.; BRETSCHNEIDER, P.; KONOTOP, I.; WESTERMANN, D.; HAGENMEYER, V.; MIKUT, R.: Virtual Storages as Theoretically Motivated Demand Response Models for Enhanced Smart Grid Operations. *Energy Technology* 4 (2016), pp. 163–176.

- [21] ZHANG, Y.; WANG, J.; WANG, X.: Review on Probabilistic Forecasting of Wind Power Generation. *Renewable and Sustainable Energy Reviews* 32 (2014), pp. 255–270.

Model Predictive Control for Learning from Demonstration

Myrel Alsayegh, Christoph Rösmann, Frank Hoffmann and
Torsten Bertram¹

¹ Lehrstuhl für Regelungssystemtechnik, TU Dortmund
E-Mail: {firstname.lastname}@tu-dortmund.de

1 Motivation and Problem Definition

Learning from Demonstration is a well established paradigm to teach robot skills in an intuitive manner. A statistical model encodes the demonstrated movements performed by the human. This model is utilized for reproduction under different situations or with diverse architectures [1]. The motion task is encoded with a Gaussian Mixture Model (GMM) which captures the probability of a data sample belonging to the underlying distribution of demonstrations.

The reproduction of the intended skill rests upon Gaussian Mixture Regression (GMR), which not only provides a reference trajectory but also captures possible variations of the task. However, the robot might not be able to exactly reproduce the designated motion due to (kino-)dynamic constraints, obstacles or disturbances. This observation demands a control scheme that reconciles the execution of the nominal task with the kinematic and dynamic capabilities of the robot and constraints imposed by the workspace. The variability encoded in the GMM provides an indicator for the model predictive control scheme how to compromise between accurate reproduction and constraints on the control effort.

Model Predictive Control (MPC) is efficient in generating an optimal trajectory with respect to an underlying objective function subject to different constraints. The combination of MPC and GMR provides a mean to attain the faithful reproduction of the skill with constraints and additional objectives of the robots motion capabilities.

The proposed approach addresses the following problem: Reproduce a motion demonstrated by a teacher in the most faithful manner by a robot which limited (kino-)dynamic capabilities prohibit a perfect imitation of the skill.

1.1 Related Work

The approach in [2] employs a minimal intervention control strategy. The controller tracks a reference motion and adapts the tracking gains based on the predicted covariances in order to reduce the control effort. The covariances reflect the relevance of the current reference signal for a faithful reproduction of the skill.

The authors in [3] propose minimal intervention control for correcting those perturbations that interfere with task performance in a temporal sense using MPC. The definition of the task performance relies on a Hidden Semi-Markov Model (HSMM) movement duration profile constructed out of the different demonstrations. Nevertheless, the authors point out the difficulty of learning this duration profile which limits the applications of this proposed model. The introduced novel MPC approach is more general as it considers the entire trajectory rather than a limited horizon.

2 Motion Reproduction in Gaussian Mixture Models with Timed Elastic Bands

The introduced approach rests upon the Timed Elastic Bands (TEB) concept originally conceived for online trajectory planning for mobile robots and further extended to model predictive control [4]. TEB generates a time optimal and collision free trajectory that is compliant with the robots kinodynamic capabilities.

The combined approach exploits the advantages of both schemes, GMM and TEB-MPC for the purpose of synthesizing trajectories that imitate the task taught by demonstration and that are compliant with the integrity condition with respect to feasible controls. The TEB-MPC allows temporal variations of the skill without the need of an explicit duration model in comparison to [3]. The spatial-temporal variability of the demonstrations is exploited in the elastic band which either adapts the path or the duration of the motion to achieve compliance with the (kino-)dynamic constraints or the disturbances.

2.1 Gaussian Mixture Models

Learning from demonstration proceeds in three consecutive phases, in the demonstration phase the motions executed by a teacher are recorded, in the learning phase a probabilistic model of the skill is generated, in the reproduction phase a generalized version of the motion is retrieved. GMM and GMR are popular approaches for learning and reproduction of motions. GMM not only capture the mean but also the variance of the motion across multiple demonstrations of the same skill.

A mixture model is able to represent multiple subpopulations within a global data distribution. It does not match individual observations with the underlying sub-population from which the sample originates.

A GMM represents the joint distribution by a composition of K normal distributions $\mathcal{N}(\cdot)$ with a set of mixture weights π_k , component means μ_k and component variance Σ_k .

In the context of learning from demonstration the GMM describes the joint probability distribution of spatial ξ_x and temporal ξ_t variables.

$$P(\xi_x, \xi_t) = \sum_k \pi_k \mathcal{N}((\xi_x, \xi_t), \mu_k, \Sigma_k) \quad (1)$$

with $\mu_k = (\mu_{x_k} \ \mu_{t_k})^T$ and $\Sigma_k = \begin{pmatrix} \Sigma_{xx_k} & \Sigma_{xt_k} \\ \Sigma_{tx_k} & \Sigma_{tt_k} \end{pmatrix}$

The parameters π_k, μ_k, Σ_k are estimated from N recorded demos $x^i(t), i \in \{1, 2, \dots, N\}, t \in [0, t_f^i]$ with iterative expectation maximization.

Figure 1 shows a set of 30 trajectories in conjunction with the probability density function of a GMM with six components estimated from the data.

The distribution of total demo time t_f^i and final states $x^i(t_f^i)$ are also characterized normal distributions

$$P(t_f) = \mathcal{N}(t_f, \mu_{t_f}, \sigma_{t_f}) \quad (2)$$

$$P(x_f) = \mathcal{N}(x_f, \mu_{x_f}, \Sigma_{x_f}) \quad (3)$$

A smoothed generalized trajectory is retrieved from the joint probability $P(\xi_x, \xi_t)$ encoded in the GMM through GMR. The generalized trajectory $P(\xi_x | \xi_t)$ denotes the mean and covariance of the expected spatial component ξ_x at time instance ξ_t .

2.2 Timed Elastic Bands

For planning of time-optimal trajectories in the context of MPC, an innovative approach based on TEB for nonlinear time-optimal model predictive control is presented in [4]. The optimal trajectory is obtained by solving an optimization problem. In contrast to conventional MPC, the TEB based MPC incorporates the time intervals of transition between consecutive discrete states explicitly as an optimization parameter.

The TEB is composed of the state and control sequences which comply with the system dynamics. The optimal control problem is formulated as a nonlinear program that is efficiently solved with online active-set or interior-point methods. The TEB approach explicitly considers the discrete time interval between consecutive states as an additional parameter subject to optimization. As the planned trajectory is *elastic* with respect to space and time its contraction and expansion w.r.t. the transition time realizes a quasi time-optimal control.

A nonlinear, autonomous dynamic system with p states and q inputs is defined by:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(t = 0) = \mathbf{x}_s \quad (4)$$

in which $\mathbf{x} \in \mathbb{R}^p$ denotes the time dependent state and $\mathbf{u} \in \mathbb{R}^q$ the corresponding control input. $\mathbf{x}_s \in \mathbb{R}^p$ denotes the initial state at time $t = 0$ s. This system of continuous differential equations is approximated by a sequence of n discrete states and $n - 1$ controls in combination with $n - 1$ strictly positive temporal discretization $\Delta T_k \in \mathbb{R}^+$.

$$\Delta T_k^{-1}(\mathbf{x}_{k+1} - \mathbf{x}_k) = \tilde{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) \quad k = 1, 2, \dots, n - 1 \quad (5)$$

The discrete states, controls and time intervals are combined into the TEB set $\mathcal{B} \subseteq \mathbb{R}^{\tilde{d}}$ with $\tilde{d} = np + (n - 1)q + 1$:

$$\mathcal{B} := \{\mathbf{x}_1, \mathbf{u}_1, \Delta T_1, \mathbf{x}_2, \mathbf{u}_2, \Delta T_2 \dots, \mathbf{x}_{n-1}, \mathbf{u}_{n-1}, \Delta T_{n-1}, \mathbf{x}_n\} \quad (6)$$

The optimization of the TEB \mathcal{B} has the objective to achieve the transition from the initial state \mathbf{x}_s to the final target state \mathbf{x}_f in minimal time T . According to Eq. (6), the transition time $T \in \mathbb{R}^+$ is determined by $T \approx (n - 1)\Delta T$. The optimal control and state sequence \mathcal{B}^* is the solution of a nonlinear program:

$$V^*(\mathcal{B}) = \min_{\mathcal{B}} \sum_k \Delta T_k \quad (7)$$

subject to

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}_s, \quad \mathbf{x}_n = \mathbf{x}_f, \quad \Delta T > 0 \\ \mathbf{h}_k(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_k, \Delta T_k) &= \mathbf{0} \quad (k = 1, 2, \dots, n-1) \\ \mathbf{g}_1(\mathbf{u}_1) &\geq \mathbf{0} \\ \mathbf{g}_k(\mathbf{x}_k, \mathbf{u}_k) &\geq \mathbf{0} \quad (k = 2, 3, \dots, n-1) \end{aligned}$$

Initial \mathbf{x}_1 and final TEB state \mathbf{x}_n are constrained by \mathbf{x}_s and \mathbf{x}_f and thus not subject to TEB optimization. The system dynamics (5) impose equality constraints $\mathbf{h}_k(\mathbf{x}_{k+1}, \mathbf{x}_k, \mathbf{u}_k, \Delta T_k) = \Delta T_k^{-1}(\mathbf{x}_{k+1} - \mathbf{x}_k) - \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{0}$. Controls and states are subject to inequality constraints $\mathbf{g}_k : \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}^r, k = 2, 3, \dots, n-1$.

In the following, the linear objective function $V(\mathcal{B}) = (n-1)\Delta T$ of the time optimal control problem, is replaced by a cross entropy that measures the Kullback-Leibler divergence $D_{KL}(q||p)$ of the TEB proposal distribution q from the GMM target distribution p . For that purpose, the TEB \mathcal{B} is mapped onto a TEB distribution q that reflects the spatio-temporal density of TEB states.

2.3 Optimal Reproduction with Timed Elastic Bands

The minimal intervention control strategy for reproduction of the skill presented in [2] proceeds in two steps

- generate a generalized reference trajectory $P(x|t)$ with mean x_d and covariance Σ_d by means of Gaussian mixture regression
- track the reference trajectory with a minimal intervention controller by means of a linear quadratic regulator (LQR) minimizing a weighted combination of tracking error and control effort

$$J = \int_{t=0}^{t_f} (\mathbf{x} - \mathbf{x}_d)^T \mathbf{Q}_d (\mathbf{x} - \mathbf{x}_d) + \mathbf{u}^T \mathbf{R} \mathbf{u} \, dt \quad (8)$$

subject to $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$. The weight matrix $\mathbf{Q}_d = \Sigma_d^{-1}$ is time dependent and inverse proportional to the covariance Σ_d . It thus reflects the variation of demonstrations inherent in the GMM. The weight matrix \mathbf{R} for the control effort is chosen such that \mathbf{u} complies with the control limits.

In that way the controller tracks a reference motion and adapts the feedback gains based on the predicted covariances in order to minimize the control effort.

The LQR-approach does not guarantee that the controls \mathbf{u} are compliant with the control limits $\mathbf{u} \in [\mathbf{u}_{\min} \ \mathbf{u}_{\max}]$. The weight matrix \mathbf{R} has to be chosen conservatively in order to avoid control signal saturation across all feasible trajectories.

Our novel approach optimizes the TEB states and controls \mathcal{B} directly w.r.t. the GMM itself rather than the generalized trajectory obtained from GMR. The key idea is that the Gaussian mixture components are considered as attractors for the states of the TEB. In other words the TEB states are supposed to cover the regions of high probability of the GMM joint distribution $P(\xi_x, \xi_t)$. For that purpose the elastic band is mapped to a probability density function via kernel density estimation. Kernel density estimation maps a finite sample set to a smoothed, non-parametric probability density function. In our case ξ_x corresponds to the TEB states \mathbf{x}_k and ξ_t to corresponding absolute time stamps $t_k = \sum_{i=1}^{k-1} \Delta T_k$.

The TEB states with time stamps (\mathbf{x}_k, t_k) are considered as independent and identically distributed samples drawn from some proposal distribution q of unknown density. The kernel density estimator of q is given by

$$q(\mathbf{x}, t) = \frac{1}{n} \sum_{i=1}^n \mathcal{N}(\mathbf{x}_k - \mathbf{x}, w_x) \mathcal{N}(t_k - t, w_t) \quad (9)$$

in which $\mathcal{N}(\mu, \sigma)$ are normal distributions, the constants w_x and w_t denote the kernel bandwidth and determine the amount of smoothing over the data.

The proposal distribution $q(\mathbf{x}, t)$ is supposed to match the target distribution $p(\mathbf{x}, t) = P(\xi_x, \xi_t)$ of the GMM. In machine learning the cross entropy defines the loss function between the true probability p_i of the true label, and the distribution q_i of the labels predicted by a classifier. The cross entropy of a proposal distribution p and a target distribution q is defined by:

$$H(p, q) = E_p[-\log q] = - \int_x p(x) \log q(x) dx \quad (10)$$

In case of a discrete sample set $X = \{x_1, x_2, \dots, x_n\}$ the cross entropy becomes

$$H(p, q) = - \sum_{x \in X} p(x) \log(q(x)) \quad (11)$$

The objective of TEB optimization is to minimize the cross entropy $H(p, q)$ between the TEB kernel density estimate and GMM target distribution

$$V^*(\mathcal{B}) = \min_{\mathcal{B}} - \sum_{x \in X} p(\mathbf{x}, t) \log(q(\mathbf{x}, t)) \quad (12)$$

in which $q(\mathbf{x}, t)$ denotes the kernel density estimate of the TEB set \mathcal{B} according to Eq. (9). In practice we generate 200 random samples \mathbf{x}, t drawn from $p(\mathbf{x}, t)$ to approximate $H(p, q)$. The optimization of the TEB is constrained by bounds on the state vector \mathbf{x} and the controls \mathbf{u} .

The nonlinear program in Eq. (12) is solved with an interior-point algorithm [6]. The interior-point approach substitutes the original problem by a sequence of approximate optimization problems. It introduces so called barrier functions with slack variables to convert the inequality constraints into equality constraints, which are easier to solve in the context of nonlinear programming. Each iteration of the approximate problem involves two steps, first a Newton step in the original parameters and slack variables to comply with the Karush-Kuhn-Tucker conditions, followed by a conjugate gradient step that employs a trust region.

The solution of the nonlinear program in Eq. (12) provides the open loop optimal control sequence. In a model predictive control scheme the initial state of the TEB \mathbf{x}_s is fed back and Eq. (12) is solved repeatedly at every control cycle. In the absence of disturbances, the closed loop solution control sequence coincides with the open loop one as the planning horizon extends towards the final state.

3 Motion Reproduction of a Parabolic Reference Profile

3.1 Trajectories and Gaussian Mixture Model

This section provides an illustrative example of reproducing a demonstrated skill under limited control effort. The demonstration data is composed of 30 parabolic trajectories which vary w.r.t. the amplitude x_p and time t_p of

the peak $x_p(t_p)$ as well as overall duration t_f and final state $x(t_f)$. Each trajectory is composed of 50 samples. The sample trajectories are shown in gray in Fig. 1.

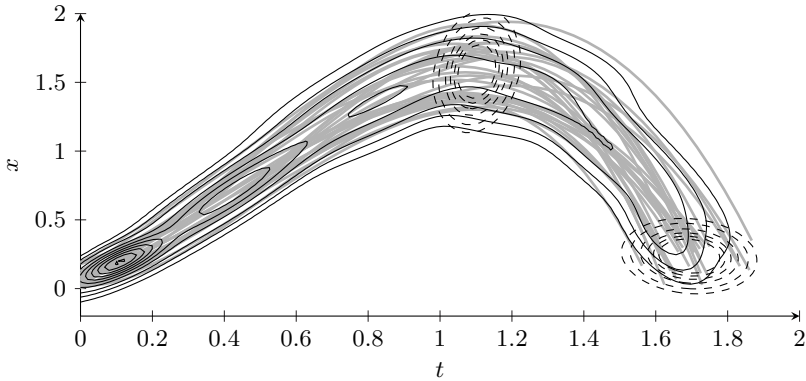


Figure 1: demonstrated trajectories (gray), probability density function of GMM (solid), probability density function of peak and final state (dashed), x and t are normed.

A GMM with six components is extracted from the set of 1500 samples (x, t) . The resulting joint probability density function $p(x, t)$ is superimposed as a contour plot in Fig. 1. The two Gaussians in Fig. 1 approximate the distribution of the peak states and the final states across 30 trajectories.

3.2 Reproduction of Motion under Dynamic Constraints

The task is the reproduction of the demonstrated trajectories in compliance with the system dynamics of a double integrator with limited controls. The double integrator system with state vector $\mathbf{x} = (x \dot{x})^T$ and control u is described by the time discrete model:

$$\begin{aligned} x_{k+1} &= x_k + \dot{x}_k \Delta T_k \\ \dot{x}_{k+1} &= \dot{x}_k + u_k \Delta T_k \end{aligned} \quad (13)$$

The state vector is constrained to $x \in [-2, 2]$ and $\dot{x} \in [-10, 10]$ the control is limited to $u \in [-u_{\max}, u_{\max}]$. The initial state at time $t = 0$ is $\mathbf{x}(0) = (x = 0, \dot{x} = 0)^T$. The constraints on the state vector do not limit the reproducing of the skill, however the control saturation imposes

a limited acceleration that does allow an accurate reproduction. A TEB with 20 states and controls is optimized to minimize the cross entropy between the kernel density estimate of the TEB and the GMM according to Eq. (12).

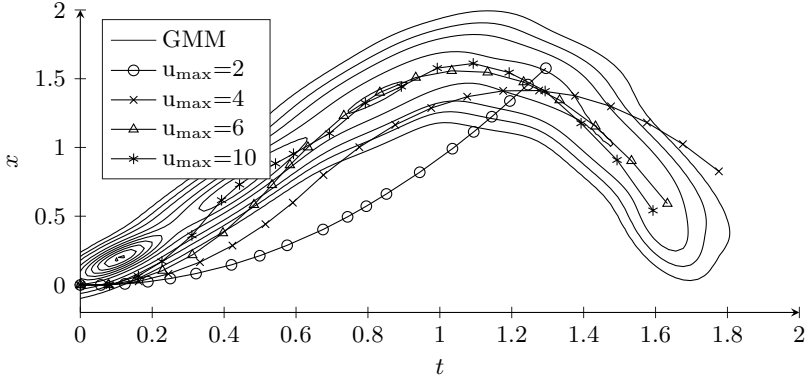


Figure 2: probability density function of GMM and optimized trajectories for control limits $u_{\max} \in \{2, 4, 6, 10\}$, x and t are normed.

Figure 2 illustrates the optimal trajectories for control saturations $u_{\max} \in \{2, 4, 6, 10\}$. None of the trajectories is able to track the initial part of the trajectory up to 0.5 seconds as the initial velocity is zero and the acceleration of the double integrator is limited. The fidelity of the reproduced trajectory increases for larger control u_{\max} , at about $u_{\max} = 6$ the integrator is able to follow the parabolic profile. Notice, that for $u_{\max} = 10$ after about $t = 0.5$ seconds the trajectory follows the maximum of the GMM probability density function and thus coincides with the mean of $P(x|t)$ obtained from GMR.

Figure 3 superimposes the probability density function of the GMM target distribution (dashed) with the proposal distribution of the kernel density estimate of the TEB states for a bandwidth $w_x = w_t = 0.2$. The figure illustrates that the TEB is shaped such that cross entropy, which up to the constant entropy $H(p)$ is equivalent to the Kullback-Leibler divergence $D_{KL}(q||p)$ of q from p , is minimized.

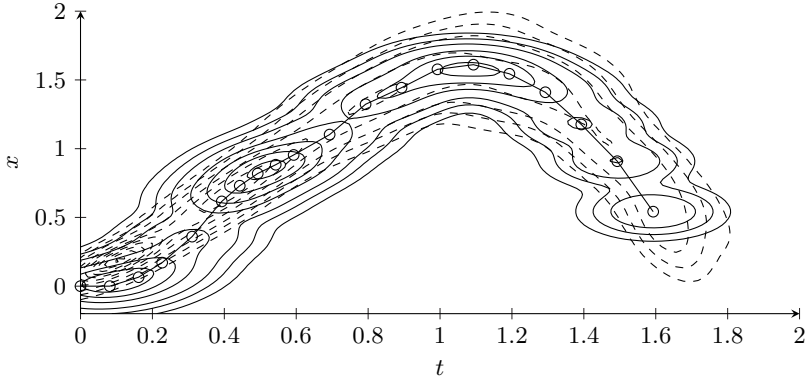


Figure 3: proposal probability density function of kernel density estimation of TEB for $u_{\max} = 10$ (solid) and target probability density function of GMM (dashed), x and t are normed.

3.3 Additional Constraints for Motion Reproduction

The concept of cross entropy minimization between proposal and target distributions is fairly general and allows the incorporation of additional constraints on the motion expressed in terms of target distributions. Such additional constraints might include the distribution of the final state or the peak state of the trajectory. Constraints either capture temporal aspects, such as maximum duration of the motion, spatial aspects such as fixed final state or spatio-temporal aspects such as a target region of the final state in time and space. The later case is illustrated in Fig. 4 which shows the normal distribution of the peak state and the final state across the 30 demonstrated trajectories in Fig. 1. The objective function for the TEB optimization in Eq. (12) is augmented by a term that captures the cross entropy between a normal distribution centered at the TEB peak respectively final state and their corresponding target distributions of the demonstrations. The control is limited to $u_{\max} = 4$ such that it is impossible for the double integrator to generate a trajectory that simultaneously traverses the peak state and reaches the target state.

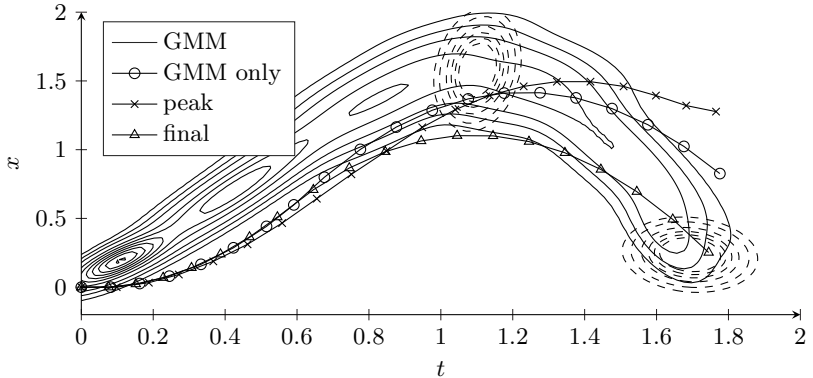


Figure 4: TEB trajectories for $u_{max} = 4$ with additional constraints imposed on peak and final state, x and t are normed

The figure shows the optimal TEB for optimization w.r.t. the GMM only, w.r.t. GMM and the peak state and w.r.t. GMM and final state. The additional constraints introduce a bias of the motion either towards compliance with the peak or final state. The approach is general and allows an online adaptation of the trajectory in order to avoid obstacles in the robot workspace.

4 Conclusions

The novel approach optimizes the trajectory w.r.t. an underlying GMM and is generally applicable to motion reproduction tasks. The model predictive approach combines the desired, faithful reproduction with feedback control to comply with dynamic constraints and respond to disturbances of the motion. The illustrative example reveals the potential of the method for model predictive motion control in the context of robot learning from demonstration.

The main drawback of the proposed scheme is the computational burden involved in the calculation of the cross entropy over a sample set. In addition, there is currently no closed form analytical expression for the gradient of the cross entropy w.r.t. TEB states and controls which requires the optimization schemes to approximate gradients with finite differences.

Currently the full solution of the nonlinear program in Eq. (12) for a TEB with 20 states requires more than a second of computation time on a desktop PC with Intel-Core-i5 processor, which limits the feasibility of the approach within a model predictive control loop. The cross entropy or corresponding Kullback-Leibler divergence of two normal distributions can be expressed in closed analytical form [5] which might provide a means for more efficient optimization of an approximative cross entropy cost function. In the future we intend to apply the scheme to reproduce more realistic skills with 6-DOF robot arms.

References

- [1] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL Press 2009.
- [2] S. Calinon, D. Bruno, D. G. Caldwell, 'A Task-Parameterized Probabilistic Model with Minimal Intervention Control' ICRA. 2014.
- [3] M. Zeestraten, S. Calinon, and D. G. Caldwell, 'Variable Duration Movement Encoding with Minimal Intervention Control' ICRA. 2016.
- [4] C. Rösmann, F. Hoffmann and T. Bertram, 'Timed-Elastic-Bands for time-optimal point-to-point nonlinear model predictive control' In: *European Control Conference (ECC), 2015*, 3352-3357. 2015.
- [5] J. Duchi, 'Derivations for Linear Algebra and Optimization', http://web.stanford.edu/~jduchi/projects/general_notes.pdf
- [6] R.H. Byrd, J. C. Gilbert, J. Nocedal, 'A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming', *Mathematical Programming*, vol. 89, no. 1, 149–185, 2000.

Robuste globale Approximatoren durch Aufspaltung der Transformationsfunktion

Jonas Schneider

Universität Osnabrück
Wachsbleiche 27, 49069 Osnabrück
Tel.: (0541) 969 2439
Fax: (0541) 969 2799
E-Mail: Jonas.Schneider@Uni-Osnabrueck.de

Kurzfassung

Designer lernfähiger Systeme benötigen bei der Wahl des Approximators Vorwissen über die Struktur des zu lernenden Problems. Dabei haben sie die Wahl zwischen einerseits robusten Approximatoren, die typischerweise durch ihre Lokalität im Eingangsraum wenig generalisieren und andererseits globalen Approximatoren, die glatt sind und stark generalisieren, jedoch oftmals wenig robust gegen Fehler in den Trainingsdaten sind und komplexere Lerngesetze benötigen. Diese Arbeit stellt einen Ansatz zur Kombination beider Approximatortypen auf Basis der Chebyshev Polynome vor, der die Robustheit von lokalen Approximatoren mit der Glattheit und Generalisierung von globalen Approximatoren kombiniert.

1 Motivation

On-Line Lernen hat in den letzten Jahren in vielen modellierungs- und automatisierungstechnischen Anwendungen Einzug gehalten, wodurch diese in der Lage sind sich im Betrieb eines technischen Systems zur Laufzeit an interne und externe Veränderungen anzupassen. Beispiele für Anwendungen sind die Modellbildung im laufenden Betrieb oder adaptive Regelungen. An dieser Stelle setzt das On-Line Lernen an, das auf dem zur Laufzeit eingehenden Strom von Daten arbeitet und im Betrieb das Modell oder die Regelung des Systems anpasst. Formal bezeichnet On-Line Lernen die Adaption der Parameter eines Approximators $\tilde{y}(x) = \varphi(x, \beta)^T \cdot \omega$

mit $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^d$ und $\omega \in \mathbb{R}^d$. φ ist eine durch β parametrisierte nicht-lineare Transformationsfunktion, die den n -dimensionalen Eingabevektor x in einen, typischerweise höherdimensionalen, Transferraum abbildet, in dem die Ausgabe über das Skalarprodukt mit ω gebildet wird. Passt das Lernen nur der Parametervektor ω an, geht der Approximator in die Form $\tilde{y} = \varphi(x) \cdot \omega$ über und wird zum LIP-Approximator (Linear in den Parametern). Diese sind durch Methoden der linearen Optimierung einfach zu trainieren und stellen daher die Grundlage in dieser Arbeit dar.

2 Approximatoren

Abhängig von der Funktion φ , bestehend aus den Basisfunktionen φ_i , lassen sich die Approximatoren in verschiedene Klassen einteilen. Auf der einen Seite gibt es die *lokalen* Basisfunktionen, wie gitterbasierte Approximatoren, bei denen pro Eingabedatum x nur eine kleine Untermenge an Basisfunktionen aktiv ist. Auf der anderen Seite gibt es *globale* Basisfunktionen, wie Polynome, bei denen jede einzelne Basisfunktion an jedem Punkt des Eingangsraums aktiv ist. Typischerweise sind lokale Approximatoren einfacher zu lernen, da es weniger Abhängigkeiten zwischen den einzelnen Parametern gibt. Somit erzielen schon Lernverfahren erster Ordnung, wie Passiv-Aggressive-Learning [4], gute Ergebnisse für lokale Approximatoren. Nachteilig ist jedoch, dass diese Approximatoren per Konstruktion keine Generalisierung über die Lerndaten hinaus bieten. Darüber hinaus sind die resultierenden Approximatoren selten glatt. Globale Approximatoren generalisieren konstruktionsbedingt bereits vom ersten Trainingsdatum an in sämtliche Bereiche des Eingangsraumes und liefern glatte Kennflächen, sind jedoch schwerer zu trainieren, so werden für solche Approximatoren oft Lernverfahren zweiter Ordnung, wie Recursive-Least-Squares (RLS), genutzt, die über eine mitgeführte Lernhistorie den Lernprozess besser kontrollieren. Nachteilig an diesen Verfahren ist die höhere Komplexität und die komplexere Parametrierung.

In dieser Arbeit wird ein neuer Ansatz zur Lokalisierung von globalen Approximatoren, hier Polynomen, vorgestellt, der die Vorteile der einfachen Lernbarkeit und Robustheit von lokalen Approximatoren mit der Glattheit und Inter- und Extrapolation von Standard-Polynomen kombiniert.

3 Konzept

Die Hauptidee des hier im Folgenden vorgestellten Konzeptes basiert auf der Aufspaltung der Transformationsfunktion $\varphi(x)$ in eine lokale Transformationsfunktion $\varphi_l(x)$ zum Lernen und eine globale Transformationsfunktion $\varphi_e(x)$ zum Auswerten. Die Idee ist somit, zunächst während des Lernens einen gitterbasierten Approximator beschrieben durch φ_l zu trainieren, um aus diesem anschließend durch φ_e durch Polynomfit einen globalen Approximator zu gewinnen. Die Idee ist somit eng mit dem Prinzip des *Lazy-Learning*[2] verwandt und verlagert die Komplexität vom Lern- in den Auswertungsschritt. Für die Funktion φ_l wird ein simpler lokaler Approximator genutzt, der jedes Trainingsdatum nur genau der nächsten Stützstelle s zuschlägt und diese in Abhängigkeit von der Entfernung des Trainingsdatums anpasst. Die Funktion φ_e stellt eine Auswertung des resultierenden Polynomfits an der Anfragestelle dar, wobei durch die Berechnung der Auswertung in Lagrange-Basis die tatsächlichen Polynomkoeffizienten nie berechnet werden müssen und Rechenaufwand eingespart wird. Die Transformationsfunktionen sind wie folgt definiert:

$$\varphi_l(x) = \begin{cases} 1 - \frac{2d(x, s_i)}{d(s_i, s_{i+1})} & \text{falls } i = \underset{j}{\operatorname{argmin}} \{d(x, s_j)\} \text{ und } \frac{d(x, s_{i+1})}{d(x, s_{i-1})} \leq 1 \\ 1 - \frac{2d(x, s_i)}{d(s_i, s_{i-1})} & \text{falls } i = \underset{j}{\operatorname{argmin}} \{d(x, s_j)\} \text{ und } \frac{d(x, s_{i+1})}{d(x, s_{i-1})} > 1 \\ 0 & \text{sonst} \end{cases} \quad (1)$$

$$\varphi_e(x) = \prod_{0 \leq m \leq n+1} \frac{x - s_m}{s - s_m} \quad (2)$$

Für die Funktion φ_l ist es notwendig die Position der Stützstellen s_i zu wählen. Das Ziel ist dabei, die Positionen so zu wählen, dass kleine Fehler in den Stützstellenhöhen nicht zu beliebig großen Fehlern in der resultierenden Approximation durch φ_e führen. Dies sollte unabhängig von der zu lernenden Funktion erfolgen, da andererseits ohne hinreichendes Vorwissen der Approximator nicht genutzt werden könnte. Die Frage ist demnach, wie sich die Fehler in den Trainingsdaten über Fehler im gitterbasierten Approximator auf die resultierende globale Polynomapproximation auswirken. Hierfür wird das Problem wie ein Standard-Polynomfit mit Berechnung der Polynomkoeffizienten $\alpha \in \mathbb{R}^m$ betrachtet. Da die Approximation direkt von den Koeffizienten des Polynoms abhängen, kann die obige Frage

umformuliert werden, wie sich Fehler in den Trainingsdaten auf die Polynomkoeffizienten auswirken. Diese können ermittelt werden durch Lösen des folgenden Gleichungssystems:

$$S \cdot \alpha = \omega \tag{3}$$

S ist dabei die Matrix bestehend aus allen Stützstellenpositionen und ω der Parametervektor der lokalen Approximation. Da es sich hierbei um ein lineares Gleichungssystem handelt, kann die Fehlerverstärkung von den Stützstellen zu den Polynomkoeffizienten durch die Kondition des Gleichungssystems abgeschätzt werden. Die Kondition ist definiert als $\text{cond}(S) = \|S\|_2 \cdot \|S^{-1}\|_2$ und nur von der Matrix S abhängig. Somit ist die Fehlerverstärkung unabhängig von der zu lernenden Funktion und nur von den Positionen der Stützstellen im Eingangsraum abhängig. Die optimalen Positionen für die Stützstellen lassen sich somit finden, wenn S wie folgt gewählt wird:

$$S = \operatorname{argmin}_S \|S\|_2 \cdot \|S^{-1}\|_2 \tag{4}$$

Durch die Matrixinversion ist dies ein komplexes Problem. Es gibt eine analytische Lösung für eindimensionale Polynome, die gegeben ist durch die Nullstellen der Chebyshev Differenzialgleichung [3]. Die resultierenden *Chebyshev Punkte* s_i können zur Designzeit berechnet werden:

$$s_i = \cos\left(\frac{2i + 1}{2(n + 1)} \cdot \pi\right), i = 1, \dots, n + 1 \tag{5}$$

Wir nennen den daraus resultierenden Approximator *Chebyshev Polynom Approximator*, der eine Orthonormalbasis für das Polynom liefert und somit die Kondition von Gleichungssystem (3) per Konstruktion minimiert. Für den zweidimensionalen Fall existieren als empirisch gefundene Lösung die *Padua-Punkte* [4]. Für höherdimensionale Polynome sind nach Wissen des Autors keine geschlossenen oder empirisch gefundenen Lösungen bekannt. Die Kondition kann jedoch durch Bildung der Basis als kartesisches Produkt der Chebyshev Punkte deutlich gegenüber einer naiven Strategie von äquidistant platzierten Stützstellen reduziert werden. Untersuchungen im mehrdimensionalen stützen dieses Vorgehen.

4 Demonstrationsbeispiel

Zur Demonstration der Leistungsfähigkeit des Approximators wird dieser mit einem Standard-Polynomapproximator, einem lokalisierten Po-

lynomapproximator mit naiv äquidistant platzierten Stützstellen, einem lokalen gitterbasierten Approximator (GLT) und einem Legendre-Polynom-Approximator verglichen. Alle Approximatoren sind vom Grad 12, haben also 13 Parameter, und werden mit dem gleichen Passive-Aggressive-Learner trainiert. Als Zielfunktion dient ein Sinus auf dem Intervall $[-1; 1]$ ohne Rauschen. Es werden 100 unabhängig und gleichverteilte Trainingsdaten genutzt. Abbildung 1 zeigt das Ergebnis der Evaluation. Im oberen Teil ist das Ergebnis der Approximation nach den 100 Trainingsdaten gezeigt. Es ist zu erkennen, dass nur der Chebyshev- und der GLT-Approximator die Funktion über den gesamten Eingangsraum wiederzugeben in der Lage sind. Das Standard- und das Legendre-Polynom sind mit dem Passive-Aggressive-Learner nicht in der Lage die Funktion wiederzugeben. Das lokalisierte Polynom mit naiv äquidistant platzierten Stützstellen zeigt starke Abweichungen von der Zielfunktion an den Rändern des Definitionsbereichs. Der untere Teil der Abbildung zeigt den Ground-Truth-Loss, also den Fehler gegen die Zielfunktion, über die Zeit. Es ist zu erkennen, dass der Chebyshev Approximator die anderen Ansätze über den gesamten Lernverlauf dominiert. Vergleichbare Ergebnisse liefert nur der GLT-Approximator, der jedoch durch die mangelnde Glattheit den Sinus an den Extrempunkten nicht wiedergeben kann. Während das Standard- und Legendre-Polynom auf Grund des Passive-Aggressive-Learners keine Verbesserung über den Verlauf erzielen, zeigt das lokale äquidistante Polynom, dass der Fehler über den Lernverlauf häufig wieder zunimmt. Dies ist auf die schlechtere Kondition des Gleichungssystems zurückzuführen, die zu starken Änderungen an den Koeffizienten des Polynoms führt. Es ist zu erkennen, dass der Chebyshev Approximator tatsächlich die gute Kondition nutzen kann und die entsprechende Stützstellenverteilung zu einem deutlich verbesserten Lernverlauf gegenüber dem Approximator mit äquidistant verteilten Stützstellen führt. Durch die Aufspaltung der Transformationsfunktion ist es darüber hinaus möglich, den Approximator mit einem Lernverfahren erster Ordnung zu trainieren. Die resultierende Approximation zeigt dabei zudem die Glattheit des Polynomapproximators. Somit schafft es der Approximator die Vorteile des lokalen gitterbasierten Approximators mit denen des globalen Polynomapproximators zu kombinieren.

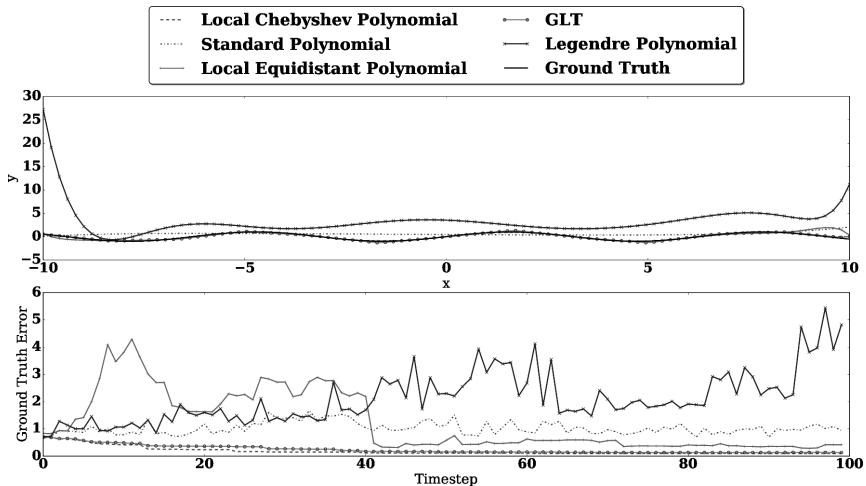


Bild 1: Oben: Vergleich des Lokalen Chebyshev-, Standard-, lokal-äquidistanten- und Legendre-Polynom-Approximator sowie GLT auf dem Sinus ohne Rauschen. Unten: Ground Truth Loss der Approximatioren.

5 Zusammenfassung

In dieser Arbeit wurde ein Ansatz für robuste und glatte Approximatoren auf Basis von Polynomen vorgestellt. Die Kernidee des Ansatzes liegt dabei in der Aufspaltung der Transformationfunktion φ in eine lokale Funktion φ_l zum Training und φ_e zur Auswertung. Durch die Lokalisierung des Polynoms unter Berücksichtigung der Konditionszahl liefert der eingeführte Chebyshev Polynom Approximator ein per Konstruktion robustes Verhalten und eine glatte Approximation. Das Demonstrationsbeispiel zeigt dieses Verhalten sogar schon für eine Zielfunktion ohne Rauschen. Der Chebyshev Polynom Approximator ist somit insbesondere geeignet eingesetzt zu werden, wenn der Designer wenig Vorwissen über die zu lernende Funktion besitzt, da er unabhängig von dieser ein robustes und glattes Verhalten zeigt.

Literatur

- [1] Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S.; Singer, Y.: Online passive-aggressive algorithms. *Journal of Machine Learning Research* 7 (2006) Mar, S. 551–585.
- [2] Aha, D. W.: *Lazy Learning*. Kap. *Lazy Learning*, S. 7–10. Norwell, MA, USA: Kluwer Academic Publishers. ISBN 0-7923-4584-3. 1997.
- [3] Mason, J. C.; Handscomb, D. C.: *Chebyshev polynomials*. CRC Press. 2002.
- [4] Bos, L.; Caliari, M.; De Marchi, S.; Vianello, M.; Xu, Y.: Bivariate Lagrange interpolation at the Padua points: the generating curve approach. *Journal of Approximation Theory* 143 (2006) 1, S. 15–25.

Skalierbare inkrementelle On-line Regression auf Simplex-Strukturen

Jan H. Schoenke

Universität Osnabrück
Wachsbleiche 27, 49069 Osnabrück
Tel.: (0541) 969 2439
Fax: (0541) 969 2799
E-Mail: Jan.Schoenke@Uni-Osnabrueck.de

1 Motivation

Die On-line Modellierung komplexer dynamischer Systeme ist ein typisches Problem im Überschneidungsbereich von Systemidentifikation und Maschinellen Lernen [1]. Zur Repräsentation der gelernten Modelle werden häufig LIP-Approximatoren (Linear In den Parametern) der Form $y(x) = \varphi(x)^T \omega$ verwendet, da sich durch die lineare Wirkung der Parameter $\omega \in \mathbb{R}^n$ auf die Ausgabe beim Lernen ein konvexes Optimierungsproblem ergibt, welches sich im laufenden Betrieb direkt lösen lässt [4, 5]. Die nicht-linearen Basisfunktionen $\varphi : \mathbb{R}^k \rightarrow \mathbb{R}^n$ bilden einen Eingabevektor $x \in \mathbb{R}^k$ aus dem Eingangsraum \mathbb{R}^k in den Parameterraum \mathbb{R}^n ab. Diese nicht-lineare Transformation φ bestimmt die Art des LIP-Approximators und dessen wesentliche Eigenschaften.

Der guten Lernbarkeit von LIP-Approximatoren steht jedoch deren begrenzte Ausdrucksstärke gegenüber, da viele Ansätze aus dieser Klasse am sogenannten *Curse of Dimensionality* (CoD - dt.: Fluch der Dimensionalität) leiden. Der CoD für LIP-Approximatoren bedeutet, dass der Aufwand für die Anwendung eines solchen Approximators hinsichtlich Rechen- und Speicherbedarf exponentiell mit der Dimension k des Eingangsraumes wächst. Ein Beispiel für einen solchen Fall sind Polynome eines bestimmten Grades d in k verschiedenen Variablen, welche sich hinsichtlich Rechen- und Speicherbedarf nach $\mathcal{O}((d+1)^k)$ verhalten. Gitterbasierte Approximatoren schaffen es durch eine Partitionierung des Eingangsraumes in Hyper-Würfel den Rechenaufwand vom Speicherbedarf zu entkoppeln, da für eine Auswertung nur zwischen den 2^k Ecken des Hyper-Würfels interpoliert wird, in welchem die aktuelle Eingabe x liegt. Aufgrund der Geometrie

des zur Interpolation verwendeten Objekts bleibt es jedoch insgesamt bei einer exponentiellen Skalierung.

Simplicial B-Splines bieten hier eine Alternative, da sie als Basisobjekt für die Interpolation Simplizes verwenden. Für *Simplicial B-Splines* 1. Ordnung existieren Algorithmen, die hinsichtlich der Rechenzeit nach $\mathcal{O}(k \log(k))$ skalieren. Diese sind zur Beschreibung dynamischer Systeme jedoch nur bedingt geeignet, da die Übergänge zwischen zwei Simplizes zwar stetig, jedoch nicht differenzierbar sind. Zusammen mit der Tatsache, dass die Anzahl der Simplizes kombinatorisch mit der Dimensionalität wächst, ergibt sich so eine sehr kantige Approximation, welche nicht zur grundsätzlichen Glattheit dynamischer Systeme passt. *Simplicial B-Splines* höherer Ordnung schaffen hier Abhilfe, indem die Nicht-Differenzierbarkeiten an den Übergängen durch Randbedingungen behoben werden. Da jedoch die Anzahl dieser Randbedingungen kombinatorisch mit der Dimensionalität wächst und diese beim Lernen des Modells zu beachten sind, lassen sich *Simplicial B-Splines* höherer Ordnung nur für niedrigdimensionale Eingangsräume verwenden, siehe [2].

2 Zielsetzung

Ziel dieser Arbeit ist es eine Interpolationsvorschrift zu entwickeln, die auch in hochdimensionalen Räumen glatt ist und hinsichtlich des Rechenaufwandes zum Auswerten und Lernen nach $\mathcal{O}(k \log(k))$ skaliert. Dazu wird im Folgenden zunächst die verwendete Geometrie der simplizialen Zerlegung des Eingangsraumes \mathbb{R}^k dargestellt und anschließend die auf dieser Topologie vorgeschlagene Interpolation beschrieben. Die Grundidee der beschriebenen Interpolationsvorschrift basiert darauf die simplizialen Komplexe, welche den Träger jeder einzelnen Basisfunktion φ_i bilden, als deformierten k -dimensionalen Hyperwürfel zu betrachten und so den aus der Geometrie von Hyperwürfeln bekannten Separationsansatz, siehe Gleichung (1), auf simpliziale Strukturen zu übertragen.

$$f(\mathbf{x}) = \prod_{i=1}^k f_i(x_i) \tag{1}$$

3 Simpliciale Zerlegung des Eingangsraumes

Die hier verwendete simpliciale Zerlegung ist eine symmetrische Variante (J1) der Coxeter-Freudenthal-Kuhn Triangulation [3]. Hierbei wird der Eingangsraum \mathbb{R}^k zunächst durch ein Gitter aus Hyperwürfeln partitioniert. Jeder einzelne dieser Hyperwürfel bildet den Rahmen für die weitere simpliciale Zerlegung, welche den Hyperwürfel in $k!$ Simplexes unterteilt. Dabei teilen sich alle so erzeugten Simplexes eine Hauptdiagonale des Würfels als längste Kante, zum Beispiel die Hauptdiagonale in Richtung $(1, 1, \dots, 1) \in \mathbb{R}^k$. Ist die Zerlegung eines Würfels durch die Wahl der gemeinsamen Hauptdiagonale festgelegt, so ergeben sich die Zerlegungen aller anderen Würfel durch Spiegelung an den $(k-1)$ -dimensionalen Hyperflächen an denen sich je zwei Hyperwürfel berühren.

In einem regelmäßigen Gitter aus m Hyperwürfeln entlang jeder Dimension gibt es bei dieser Zerlegung insgesamt $k! \cdot m^k$ Simplexes. Um aus diesen den Simplex auszuwählen, welcher die aktuelle Eingabe \mathbf{x} enthält, muss zunächst der Hyperwürfel H_p bestimmt werden in den diese Eingabe fällt. Anschließend liefert die Sortierung der normierten Hyperwürfel-internen Koordinaten $\mathbf{u}_p(\mathbf{x}) : \mathbb{R}^k \rightarrow [0, 1]^k$ der Eingabe \mathbf{x} den gesuchten Simplex. Für die Betrachtungen im Rahmen dieser Arbeit genügt es sich auf die Zerlegung eines Hyperwürfels H_p zu beschränken, weshalb hier im Folgenden allgemeine interne Koordinaten $\mathbf{u} \in [0, 1]^k$ betrachtet werden ohne Bezug zu einem bestimmten Hyperwürfel zu nehmen. Innerhalb dieses Hyperwürfels werden die Ecken der darin beschriebenen Simplexes durch normierte Koordinaten $s_i \in [0, 1]^k$ beschrieben, wobei $(0, 0, \dots, 0) \in [0, 1]^k$ mit dem Beginn der Hauptdiagonale des Würfels identifiziert wird und $(1, 1, \dots, 1) \in [0, 1]^k$ mit deren Ende. Die Sortierung der internen Koordinaten sei mit $P = (p_1, p_2, \dots, p_k) \in \mathbb{N}^k$ beschrieben, sodass gilt:

$$u_{p_1} \geq u_{p_2} \geq \dots \geq u_{p_k} \tag{2}$$

Läuft man in der Reihenfolge P entlang der Kanten des Hyperwürfels $\{0, 1\}^k$ von $\mathbf{0}$ nach $\mathbf{1}$ so ergeben sich aus den besuchten Ecken gerade die normierten Koordinaten s_0, s_1, \dots, s_k des zugehörigen Simplex. Für die Auswahl des zur Eingabe \mathbf{u} gehörigen Simplex genügt es somit dessen Komponenten zu sortieren, woraus sich ein Rechenaufwand nach $\mathcal{O}(k \log(k))$ ergibt.

4 Interpolation auf simplizialen Strukturen

Die Basisfunktionen φ_i des gesuchten LIP-Approximators werden für die einzelnen Eckpunkte s_i der Simplizes definiert. Dabei erstreckt sich jede Basisfunktion φ_i je nach ihrer Nähe zu $\mathbf{0}$ beziehungsweise $\mathbf{1}$ über unterschiedlich viele Simplizes. Um den Separationsansatz aus der Hyperwürfelgeometrie auf diese Basisfunktionen übertragen zu können, muss die Geometrie der simplizialen Komplexe der Basisfunktionen berücksichtigt werden. Während für jeden einzelnen Simplex die Ordnung aus Gleichung (2) gilt, so vereinfacht sich diese für den simplizialen Komplex eines Eckpunktes s_i zu:

$$\min\{u_{p_1}, u_{p_2}, \dots, u_{p_i}\} \geq \max\{u_{p_{i+1}}, u_{p_{i+2}}, \dots, u_{p_k}\} \quad (3)$$

$$\Rightarrow \quad u_{p_i} \geq u_{p_{i+1}} \quad (4)$$

Die Ordnung der betrachteten simplizialen Komplexe aus Gleichung (3) unterscheidet somit nur zwei Gruppen von Variablen, die sich jeweils gegenseitig über ihr Minimum beziehungsweise Maximum beschränken. Gekoppelt sind diese beiden Gruppen allein durch Gleichung (4), welche der Geometrie eines 2D-Simplex entspricht und für die Anwendung des Separationsansatzes berücksichtigt werden muss. Für s_0 und s_k ist jeweils eine der beiden Gruppen leer, da diese beiden Eckpunkte von allen Simplizes des Würfel geteilt werden und sich somit direkt die Geometrie eines Hyperwürfels ergibt.

Mittels des Separationsansatzes aus Gleichung (1) lassen sich die Eigenschaften einer eindimensionalen Funktion direkt auf den mehrdimensionalen Fall übertragen. Für eine differenzierbare Interpolation auf dem Einheitsintervall bietet sich zum Beispiel das Polynom $g(x)$ aus Gleichung (5) an.

$$g(x) = 2x^3 - 3x^2 + 1 \quad (5)$$

Aufgrund der Symmetrie der Zerlegung des Eingangsraumes genügt es die Basisfunktionen $\varphi_i(\mathbf{u})$ für $i = 1, 2, \dots, k-1$ zu betrachten, die zu den internen Eckpunkten s_1, s_2, \dots, s_{k-1} eines Simplex gehören und wie folgt zu berechnen sind:

$$\begin{aligned} \varphi_i(\mathbf{u}) &= \frac{g(1-u_{p_i})}{g(1-u_{p_i}) + g(u_{p_{i+1}})} \prod_{j=1}^i g(1-u_{p_j}) \prod_{j=i+1}^k g\left(\frac{u_{p_j}}{u_{p_i}}\right) \quad (6) \\ &+ \frac{g(u_{p_{i+1}})}{g(1-u_{p_i}) + g(u_{p_{i+1}})} \prod_{j=1}^i g\left(\frac{1-u_{p_j}}{1-u_{p_{i+1}}}\right) \prod_{j=i+1}^k g(u_{p_j}) \end{aligned}$$

Bei der in Gleichung (6) beschriebenen Interpolation zwischen den beiden Produkttermen kann der Nenner null werden. In diesem Fall liegt \mathbf{u} genau auf s_i und φ_i kann stetig differenzierbar fortgesetzt werden. Für das Training des durch die φ_i beschriebenen LIP-Approximators lassen sich Standard On-line Lernverfahren wie [4, 5] verwenden.

5 Zusammenfassung

In dieser Arbeit wurde eine differenzierbare multivariate Interpolationsvorschrift für eine LIP-Approximation auf einer symmetrischen simplizialen Struktur vorgestellt, die hinsichtlich des Rechenaufwandes nach $\mathcal{O}(k \log(k))$ skaliert, wobei k die Dimension des Eingangsraumes angibt über dem approximiert wird. Grundlage der Interpolationsvorschrift ist die Übertragung des Separationsansatzes für hypercubische Strukturen auf symmetrische simpliziale Strukturen.

Literatur

- [1] Sigaud, O.; Salaün, C.; Padois, V.: On-line regression algorithms for learning mechanical models of robots: a survey. *Robotics and Autonomous Systems* 59 (2011) 12, S. 1115–1129.
- [2] Sun, L. G.: *Model and Sensor Based Nonlinear Adaptive Flight Control with Online System Identification*. TU Delft, Delft University of Technology. 2014.
- [3] Moore, D. W.: *Simplicial mesh generation with applications*. Techn. Ber., Cornell University. 1992.
- [4] Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S.; Singer, Y.: Online passive-aggressive algorithms. *Journal of Machine Learning Research* 7 (2006) Mar, S. 551–585.
- [5] Crammer, K.; Lee, D. D.: Learning via gaussian herding. In: *Advances in neural information processing systems*, S. 451–459. 2010.

Strukturanalyse eines neuronalen Netzes zur Spurwechselprädiktion für das automatisierte Fahren

Martin Krüger^{1,2}, Stefan Meuresch¹, Anne Stockem Novo¹, Till Nattermann¹, Karl-Heinz Glander¹, Torsten Bertram²

¹ZF TRW, Techcenter Düsseldorf
E-Mail: martin.krueger@zf.com

²Lehrstuhl für Regelungssystemtechnik, TU Dortmund

1 Einleitung

Automatisierte Fahrfunktionen können durch die Verlagerung der Fahraufgabe weg vom menschlichen Fahrer, als Hauptursache für Verkehrsunfälle [1], zur Steigerung der Sicherheit im Straßenverkehr beitragen. Durch die Optimierung von Fahrstrecke und Fahrdauer, sowie eine bessere Ausnutzung des zur Verfügung stehenden Verkehrsraumes [2], kann das automatisierte Fahren auch zur Effizienzsteigerung des Verkehrs beitragen.

Besonders Fahrfunktionen, die SAE Level 3 oder höher [3] zuzuordnen sind, werden in ihrem Einsatz komplexen Verkehrssituationen ausgesetzt sein. Das bedeutet, dass diese Systeme nicht nur die das Ego-Fahrzeug umgebenden statischen und dynamischen Objekte zuverlässig erkennen müssen, sie müssen darüber hinaus auch in der Lage sein, das menschliche Verhalten anderer Verkehrsteilnehmer angemessen zu berücksichtigen. Das Fahrverhalten menschlicher Fahrer weist eine sehr große Varianz auf, da es nicht nur durch den persönlichen Fahrstil [4] bestimmt wird, sondern darüber hinaus auch vom aktuellen mentalen Zustand des Fahrers abhängt [5], [6]. Dieser Umstand trägt zu einer hohen Komplexität der Fahrerintentionserkennung bei. Solche Systeme können nur schwer durch klassische regelbasierte Verfahren realisiert werden. Dies wird noch verstärkt dadurch, dass es sehr schwer bis unmöglich erscheint, vorab alle möglicherweise auftretenden Situationen zu beschreiben. Künstliche neuronale Netze als Methode aus dem Gebiet des maschinellen Lernens zeichnen sich dadurch aus, dass sie Muster aus vorgegebenen Daten lernen und verallgemeinern können.

Da sich Autobahnen durch eine vergleichsweise geringe Varianz ihrer Erscheinungsformen gegenüber urbanen Situationen auszeichnen und ihre Komplexität durch die Abwesenheit von Querverkehr geringer ausfällt, konzentrieren sich automatisierte Fahrfunktionen bisher vorrangig auf diesen Straßentyp. Ein elementares Fahrmanöver auf Autobahnen ist der Spurwechsel. Spurwechsel anderer Verkehrsteilnehmer beeinflussen durch die Veränderung der Spurbelegung die Fahrzeugführung des Ego-Fahrzeuges. Je früher Informationen über beabsichtigte Spurwechsel von Umgebungsfahrzeugen verfügbar sind, umso eher kann das Ego-Fahrzeug darauf reagieren und um so geringer können notwendige Anpassungen des Fahrzustandes des Ego-Fahrzeuges ausfallen. Für die möglichst frühzeitige Erkennung von Spurwechselmanövern auf Autobahnen eignen sich neuronale Netze auf Grund ihrer Generalisierungsfähigkeit bezüglich des Fahrverhaltens anderer Verkehrsteilnehmer. Dieser Beitrag stellt eine Methode zur Spurwechselerkennung des Ego-Fahrzeuges aus Simulationsdaten vor. Das Verfahren soll zukünftig auf Umgebungsverkehr und die Nutzung von Realdaten erweitern werden. Weiterhin zeigt die vorliegende Arbeit Vorgehensweisen auf, um die Anzahl der Netzwerkparameter schrittweise zu optimieren.

2 Relevante Arbeiten

Die frühzeitige Erkennung der Spurwechselabsicht ist schon seit einiger Zeit Gegenstand von Untersuchungen. Im Gegensatz zur vorliegenden Arbeit, die einen Beitrag zur Spurwechselerkennung anderer Verkehrsteilnehmer leisten möchte, befasst sich die aufgeführte Literatur meist mit der Vorhersage von Spurwechseln des Ego-Fahrzeuges, wenn nicht explizit anders angegeben (beispielsweise [7]).

Für die praktische Umsetzung wurden im Laufe der Zeit verschiedenste Methoden entwickelt. In [15] wird beispielsweise ein System zur Erkennung von Spurwechseln mit Hidden Markov Modellen vorgestellt. Die im Folgenden angestellten Betrachtungen sollen sich jedoch besonders auf die Methoden im Bereich Support Vector Machines (SVM) und Computational Networks (CN) konzentrieren. [7] nutzt Bayes'sche Netze zur Spurwechselerkennung der umgebenden Verkehrsteilnehmer. Die Arbeit verwendet als Eingangsdaten den lateralen Abstand des Zielfahrzeuges von der Fahrspurmitte, die laterale Geschwindigkeit, die Zeitlücke auf der Nachbarspur sowie Fahrtrajektorien-bezogene Daten aus aufgenommenen Realdaten. In [7] wird angegeben, dass Spurwechsel durchschnittlich 1,05 s vor dem

tatsächlichen Spurwechsel erkannt werden können. [11] verwendet ebenfalls Bayes'sche Netze um Spurwechsel des Umgebungsverkehrs auf Realdaten zu präzisieren. Die Qualität des Ansatzes wird jedoch nicht absolut angegeben, sondern als Vergleich zu einem herkömmlichen Abstandstempomatsystem, wobei die Manöver mit dem vorgestellten Ansatz durchschnittlich 0,6 s früher erkannt werden können.

In [8] wird ein Verfahren zur Fahrspurwechsellerkennung mittels SVM beschrieben. Zum Training werden Realdaten mit folgenden Merkmalen verwendet: Geschwindigkeit, Lenkradwinkel, Beschleunigung (lateral und longitudinal), lateraler Abstand zur Spurmitte und Abstand zum Vorderfahrzeug. Durchschnittlich 0,3 s nach Beginn des Spurwechsels kann dieser mit 87 % Wahrscheinlichkeit prognostiziert werden. In dieser Arbeit wird besonderer Wert auf die Untersuchung des Einflusses des Merkmals lateraler Abstand zur Spurmitte gelegt. Eine weitere Besonderheit stellt die Berechnung der Merkmale dar. Die Eingangsgrößen werden nicht selbst als Merkmale verwendet, sondern die Varianz dieser Größen über ein gewisses Zeitfenster wird als Eingang genutzt.

Ein häufig auftretender Ansatz zur Spurwechsellerkennung ist die Kombination von SVM mit Bayes'schen Netzen [10], [11], [12] oder [13]. [10] präsentiert einen sogenannten ganzheitlichen Ansatz, was bedeutet, dass er eine weite Spanne an Merkmalen zur Auswertung des Erkennungsverfahrens nutzt, wie beispielsweise Fahrdynamikgrößen, Ausgangsgrößen der Seit- und des Front-Radars, Fahrspurkameradaten und sogar Daten einer Kamera, die den Kopf des Fahrers erfasst. Damit werden laut der Arbeit Prädiktionszeiten von bis zu 3 s erreicht. Es wird jedoch auch auf eine schlechter werdende Leistungsfähigkeit hingewiesen, sobald das System im Fahrzeug in Echtzeit betrieben wird. Die Arbeit von [13] konzentriert sich darauf, mit einer SVM und einem Bayes'schen Netzwerk Spurwechsel in Realdaten, basierend auf den Merkmalen lateraler Abstand zur Spurmitte und Lenkwinkel, möglichst sicher 1,2 s vor dem tatsächlichen Spurwechsel zu erkennen. In der Arbeit von [12] wird ebenfalls eine Kombination aus SVM und Bayes'schen Methoden zur Spurwechsellerkennung des Ego-Fahrzeuges verwendet. An Hand der Merkmale lateraler Abstand zur Spurmitte, laterale Geschwindigkeit, Lenkwinkel und Lenkwinkelgeschwindigkeit können Spurwechsel in Realdaten durchschnittlich 1,3 s vor dem tatsächlichen Ereignis erkannt werden.

Die Arbeit [9] vergleicht sowohl verschiedene Techniken des maschinellen Lernens zur Spurwechsellerkennung: SVM, vorwärts gerichtete neuronale

Netze (Feedforward Neural Networks – FFNN) sowie rekurrente neuronale Netze (Recurrent Neural Networks – RNN) und evaluiert darüber hinaus zusätzlich die geeignetste Kombination an Eingangsmerkmalen zur Spurwechselprädiktion. Zunächst wird mit einem festen Satz an Merkmalen, der die Größen lateraler Abstand zur Spurmitte, laterale Geschwindigkeit, laterale Beschleunigung, Lenkradwinkel, Kurvenradius und Time To Collision (TTC) umfasst, die SVM als die geeignetste Methode bestimmt (Erkennung des Spurwechsels auf Simulationsdaten durchschnittlich 1,62 s vor dem tatsächlichen Spurwechsel). Anschließende Untersuchungen mit diesem Verfahren ergeben, dass die Kombination der Größen lateraler Abstand zur Spurmitte, Lenkwinkel, TTC und gegebenenfalls der Kurvenradius als besonders relevant für die Spurwechselerkennung anzusehen ist.

In [14] werden mit Hilfe eines naiven Bayes Klassifikators aus einer Vielzahl an möglichen Spurwechselerkennungsmerkmalen die Eingangsdaten bestimmt, die am stärksten zur Unterscheidbarkeit der Zustände kein Spurwechsel, Spurwechsel links und Spurwechsel rechts beitragen. In dieser Arbeit wird betont, dass die Daten in den drei zu erkennenden Klassen ungleich verteilt sind. Die Arbeit identifiziert die Merkmale lateraler Abstand zur Spurmitte, laterale Geschwindigkeit und Relativgeschwindigkeit zum vorausfahrenden Fahrzeug als die zur Manöverdetektion für das Umgebungsfahrzeug relevantesten Eingangsdaten. Spurwechsel können bis zu 2,2 s vor dem tatsächlichen Spurwechsel erkannt werden.

Die zusammengetragenen Ergebnisse sind untereinander und mit den in dieser Arbeit präsentierten Resultaten oft nur schwer vergleichbar, da sich die Definitionen des Manövers Spurwechsel teilweise erheblich voneinander unterscheiden.

3 Feedforward Network

Neuronale Netze als Methode des maschinellen Lernens sind der Datenverarbeitung im zentralen Nervensystem des Menschen nachempfunden, speziell dem Gehirn. Eingehende Daten werden von den elementaren Einheiten der neuronalen Netze (künstliches neuronales Netz: künstliches Neuron; natürliches neuronales Netz: natürliches Neuron) empfangen, verarbeitet und anschließend als Eingangsdaten an das nächste Neuron weitergeleitet. Anders als im Gehirn sind die Neuronen künstlicher neuronaler Netze meist in Schichten streng strukturiert angeordnet. Dem natürlichen Vorbild und

der künstlichen Nachbildung gemeinsam ist, dass die Stärke des Ausgangssignals eines Neurons von der Stärke des Neuroneneingangs und der Art der Aktivierung des Neurons abhängt. Wissen wird in beiden Systemen somit nicht konzentriert an bestimmten Punkten gespeichert, sondern wird durch viele, das Gesamtsystem beschreibende und verteilte Netzparameter abgebildet. Diese Netzwerkparameter werden bei künstlichen neuronalen Netzen während der Trainingsphase so optimiert, dass der resultierende Netzausgang möglichst gut mit einem vorgegebenen Sollwert übereinstimmt (überwachtes Lernen). Davon prinzipiell zu unterscheiden ist die sich an das Training anschließende Ausführungsphase des Netzes. Während dieser Phase soll das Netz in seinem beabsichtigten Einsatzfall zur Datenverarbeitung genutzt werden, ohne dass Anpassungen der Netzwerkparameter stattfinden. Eine spezielle Art künstlicher neuronaler Netze ist das FFNN, bei dem Daten stets nur in Richtung des Ausgangs übertragen werden. Das Multilayer Perceptron (MLP) stellt wiederum eine besondere Form von FFNN dar. Auf den Aufbau und die Funktionsweise dieses Typs soll in den folgenden Abschnitten eingegangen werden.

3.1 Neuronenschichten

Neuronale Netze sind typischerweise hierarchisch in Schichten aufgebaut. Dabei ist der Ausgang o_i^l des i -ten Neurons der Schicht l ein Eingang x_i^{l+1} des j -ten Neurons der Schicht $l + 1$: $o_i^l = x_i^{l+1}$. Das vorgelagerte oder das das neuronale Netz einbettende System übergibt die Daten f_i , die vom Netz verarbeitet werden sollen an die Eingangsschicht des Netzes (Input Layer). In dieser Schicht werden die Eingangsdaten bei der Betrachtung von Zeitreihen-Problemen typischerweise mittels Z-Transformation (1) normiert [21]:

$$o_{i, NormIn}^1 = \frac{f_i - \bar{f}}{\sigma} \quad (1)$$

Die Größen \bar{f} und σ werden einmalig während des Trainings des neuronalen Netzes aus den Trainingsdaten bestimmt. Das MLP zeichnet sich dadurch aus, dass es vollständig kontaktiert ist. Das bedeutet, dass der Ausgang eines Neurons allen Neuronen der nachfolgenden Schicht als Eingang zur Verfügung steht. Alle Neuronen, die nicht zur Eingangsschicht gehören, können somit mehrere Eingänge haben. Die Verbindung eines Neurons mit einem anderen wird durch ein sogenanntes Gewicht w_{ji} charakterisiert,

welches angibt, wie groß der Einfluss des Ausgangs des vorherigen Neurons i (der Schicht $l - 1$) auf das nachfolgende Neuron j (der Schicht l) sein soll. Der Neuroneneingang z_j der Schicht l ergibt sich dann als Summe der gewichteten Dateneingänge o_i über alle n Neuronen der Schicht $l - 1$ und des Bias b_j , der als Schwellwert zur Aktivierung des Neurons interpretiert werden kann [17], siehe (2). Der Bias b_j berechnet sich als Produkt aus der konstanten Aktivierung des Biasneurons mit 1 und dem Biasgewicht w_{0j} als zu optimierendem Netzwerkparameter. Auf Grund der konstanten Aktivierung mit 1, entspricht der Bias, stets dem Biasgewicht. Der Bias selbst kann somit ebenfalls als Netzwerkparameter angesehen werden:

$$z_j^l = \sum_{i=1}^n (o_i^{l-1} w_{ji}^l) + b_j^l \quad (2)$$

Der Ausgang eines Neurons o_j^l ergibt sich durch die Anwendung der Aktivierungsfunktion auf den Neuroneneingang z_j^l . Für Klassifikations- und Regressionsaufgaben wird als Aktivierungsfunktion der Ausgangsschicht L des Netzes (Output Layer) häufig die softmax-Funktion verwendet, die die Summe aller Netzwerkausgänge y_j (Ausgang y des j -ten Neurons der Ausgangsschicht: $y_j = x_j^L$) auf 1 beziehungsweise 100 % normiert (3). Schichten die zwischen der Eingangs- und Ausgangsschicht liegen werden als verdeckte Schichten (Hidden Layer) bezeichnet. Als Aktivierungsfunktion dieser Schichten werden meist sigmoide Funktionen (wie die tanh- oder die logistische Sigmoid-Funktion) oder die ReLU-Funktion (Rectified Linear Unit) gewählt (4):

$$y_j = \frac{e^{z_j^L}}{\sum_{i=1}^n (e^{z_i^L})} \quad (3)$$

$$o_j^l = \tanh(z_j^l) \quad (4)$$

3.2 Parameteroptimierung beim überwachten Lernen

Die Optimierung der Netzwerkparameter (Gewichte und Bias) während der Trainingsphase unterteilt sich in zwei Abschnitte, den sogenannten Forward Pass, die Verarbeitung der Eingangsdaten in Richtung des Netzausganges und den Backward Pass, während dessen der am Ausgang ermittelte Fehler

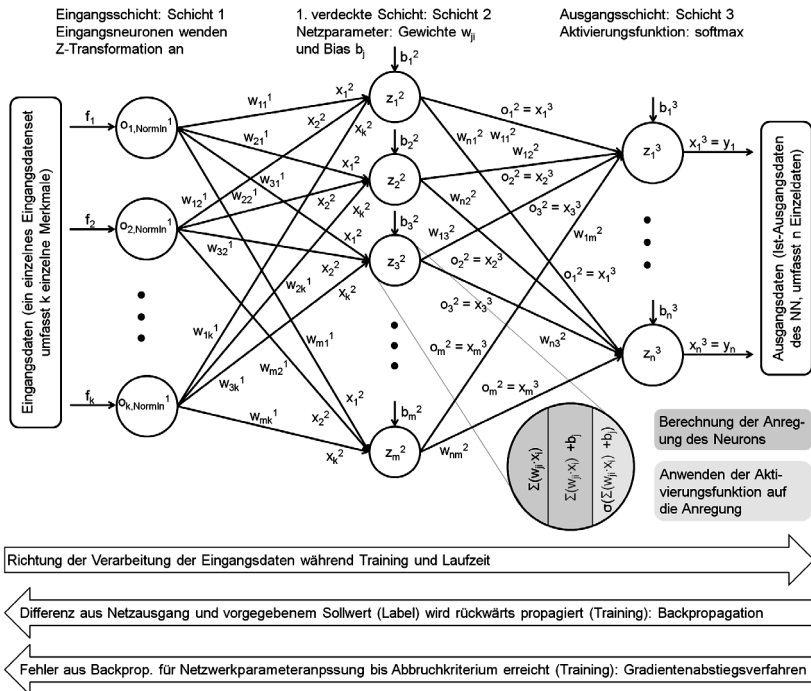


Bild 1: Darstellung des Trainingsprozesses eines MLP

rückwärts durch das Netz propagiert wird und daran anschließend die Netzwerkparameter dementsprechend angepasst werden. Beim Forward Pass erfolgt die Verarbeitung der Trainingsdaten in einzelnen Datenpaketen, schichtweise bis zum Netzwerkausgang. Die prinzipielle Vorgehensweise während des Trainings eines MLP ist in Abbildung 1 dargestellt.

Beim Training neuronaler Netze wird zwischen dem Online-Training, bei dem die Gewichte und die Bias nach der Berechnung jedes einzelnen Datenpaketes angepasst werden, und dem Batch-Training, bei dem die Netzwerkparameter nach Berechnungen von p Datenpaketen (den sogenannten Batches) angepasst werden, unterschieden. Da beim überwachten Lernen der gewünschte Soll-Ausgangswert des Netzes bekannt ist, kann der Fehler am Netzwerkausgang δ_j^L als Differenz zwischen Soll- und Ist-Ausgangswert t_j und y_j angegeben werden (5). Mit Hilfe des Backpropagation Algorithmus kann der Fehler am Netzwerkausgang δ_j^L zurückgerechnet werden,

sodass die Abweichungen in den gegebenenfalls vorhandenen verdeckten Schichten δ_j^l berechnet werden können (6). Dazu ist es notwendig die Ableitung der Aktivierungsfunktion (softmax' beziehungsweise tanh') an der Stelle der jeweiligen Aktivierung des Neurons z_j^l auszuwerten [18]:

$$\delta_j^L = \text{softmax}'(z_j^L)(y_j - t_j) \quad (5)$$

$$\delta_j^l = \text{tanh}'(z_j^l) \sum_k (\delta_k^{l+1} w_{kj}^{l+1}) \quad (6)$$

Die Gewichte des neuronalen Netzes werden anschließend wie folgt angepasst: $w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}$. Das als Fehler- oder Kostenfunktion bezeichnete Optimierungskriterium C (mathematische Berechnung meist als Cross Entropy oder Summe der Fehlerquadrate) wird dadurch epochenweise kontinuierlich minimiert. Je nachdem ob ein Gewicht der Ausgangsschicht (7) oder einer verdeckten Schicht (8) angepasst wird, ergibt sich die Änderung der Gewichte Δw , nach [18], zu:

$$\Delta w_{ji}^L = -\eta x_i^l \text{softmax}'(z_j^L)(y_j - t_j) \quad (7)$$

$$\Delta w_{ji}^l = -\eta x_i^l \text{tanh}'(z_j^l) \sum_k (\delta_k^{l+1} w_{kj}^{l+1}) \quad (8)$$

Die Lernrate η gibt an wie schnell das neuronale Netz bei einem gegebenen Fehler lernen soll, beziehungsweise wie stark die Anpassung der Gewichte ausfällt. Die Optimierung der Netzwerkparameter erfolgt epochenweise so lange bis ein Abbruchkriterium erfüllt ist (beispielsweise das Erreichen einer zuvor festgelegten Anzahl an Epochen oder das Unterschreiten eines Grenzwertes der Kostenfunktion).

4 Ausgangssituation bezüglich der Spurwechselerkennung

Aufgabe der Arbeit von [20] ist es, ein neuronales Netz zur Spurwechselerkennung zu entwickeln. Die Idee besteht darin neuronale Netze auf Grundlage der Simulationsdaten von [19] zu trainieren und geeignete trainierte Netze anschließend in einem realen Versuchsfahrzeug zu testen. Dieses Vorgehen

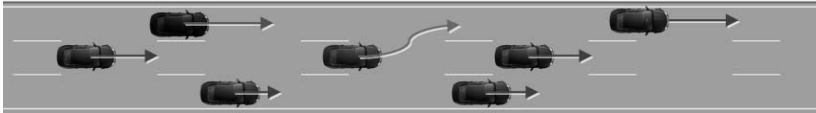


Bild 2: Autobahnsituation und Spurwechselabsicht des mittleren Fahrzeuges

umfasst eine weitere Abstraktionsstufe, da die Simulationsdaten, die zum Training verwendet werden, das Verhalten des Ego-Fahrzeuges darstellen, im Versuchsfahrzeug jedoch Spurwechsel von umgebenden Verkehrsteilnehmern prädiziert werden sollen. Bezogen auf Abbildung 2 bedeutet dies, dass sich die Simulationsdaten auf das mittlere Fahrzeug beziehen, welches beabsichtigt einen Spurwechsel nach links durchzuführen, wenn es die Verkehrssituation erlaubt. Im Versuchsfahrzeug sollen später Spurwechsel des mittleren Fahrzeuges erkannt werden, wenn beispielsweise das Ego-Fahrzeug die Rolle des verfolgenden Fahrzeuges auf der linken Spur einnimmt.

Ein Spurwechsel wird in dieser Arbeit definiert als die Fahrzeugbewegung ab dem Zeitpunkt zu dem das Fahrzeug beginnt sich vor dem eigentlichen Spurwechsel mit stets positiver beziehungsweise negativer lateraler Geschwindigkeit in Richtung der gewünschten Nachbarspur zu bewegen (eine Bewegung des Fahrzeugs mit positiver Geschwindigkeit führt zu einem Spurwechsel nach rechts, eine Bewegung mit negativer Geschwindigkeit zu einem Spurwechsel nach links). Der Spurwechsel endet laut Definition sobald der geometrische Mittelpunkt des Fahrzeuges die Fahrspurmarkierung überschritten hat. Entsprechend dieser Definition erfolgt das Labeln der Daten in [20].

Die Arbeit [20] verwendet, wie viele der im Abschnitt 2 vorgestellten Beiträge, Merkmale als Netzwerkeingangsdaten, die sich in zwei Gruppen einteilen lassen. Die Größen lateraler Abstand zur Spurmitte, laterale Geschwindigkeit und die Absolutgeschwindigkeit beziehen sich auf das Fahrverhalten des betrachteten Fahrzeuges. Die Relativgeschwindigkeiten und Abstände des zu betrachtenden Fahrzeuges zu den Verkehrsteilnehmern des Umgebungsverkehrs bilden die Umgebungssituation ab und sollen angeben, ob ein Spurwechsel aus der Verkehrssituation heraus motiviert ist. Tabelle 1 stellt die genutzten Merkmale übersichtlich zusammen.

Da die vorliegende Problemstellung als Analyse von Zeitreihen betrachtet werden kann, untersucht [20] verschiedene Möglichkeiten der Berücksichtigung des zeitlichen Verlaufs der Eingangsgrößen. Zum einen werden FFNN

Tabelle 1: Übersicht der in [20] verwendeten Netzeingangsdaten

Eingangsgröße	Bedeutung
d_{lat}	Lateraler Abstand zur Spurmitte
v_{lat}	Laterale Geschwindigkeit
v	Absolutgeschwindigkeit
$d_{long, ahead-left}$	Longitudinaler Abstand zum vorausfahrenden Fahrzeug auf der linken Nachbarspur
$\Delta v_{long, ahead-left}$	Longitudinale Differenzgeschwindigkeit zum vorausfahrenden Fahrzeug auf der linken Nachbarspur
$d_{long, ahead-central}$	Longitudinaler Abstand zum vorausfahrenden Fahrzeug auf der eigenen Spur
$\Delta v_{long, ahead-central}$	Longitudinale Differenzgeschwindigkeit zum vorausfahrenden Fahrzeug auf der eigenen Spur
$d_{long, ahead-right}$	Longitudinaler Abstand zum vorausfahrenden Fahrzeug auf der rechten Nachbarspur
$\Delta v_{long, ahead-right}$	Longitudinale Differenzgeschwindigkeit zum vorausfahrenden Fahrzeug auf der rechten Nachbarspur
$d_{long, behind-left}$	Longitudinaler Abstand zum hinterherfahrenden Fahrzeug auf der linken Nachbarspur
$\Delta v_{long, behind-left}$	Longitudinale Differenzgeschwindigkeit zum hinterherfahrenden Fahrzeug auf der linken Nachbarspur
$d_{long, behind-central}$	Longitudinaler Abstand zum hinterherfahrenden Fahrzeug auf der eigenen Spur
$\Delta v_{long, behind-central}$	Longitudinale Differenzgeschwindigkeit zum hinterherfahrenden Fahrzeug auf der eigenen Spur
$d_{long, behind-right}$	Longitudinaler Abstand zum hinterherfahrenden Fahrzeug auf der rechten Nachbarspur
$\Delta v_{long, behind-right}$	Longitudinale Differenzgeschwindigkeit zum hinterherfahrenden Fahrzeug auf der rechten Nachbarspur

verwendet, denen neben den aktuellen Eingangswerten auch Daten vorangegangener Zeitschritte als Historie der Merkmale zur Verfügung gestellt werden. Je Eingangsgröße werden neun weitere Datenpunkte dieses Merkmals, die um je 0,1 s weiter zurück liegen, als Historie gewählt. Sowohl die Anzahl als auch die Zeitdifferenz der zusätzlichen Werte werden in [20] festgelegt und sind somit nicht Gegenstand weiterer Untersuchungen. Dieses Vorgehen wird gewählt, um die einfache Struktur von FFNN und die zusätzlichen Informationen zur Zeitreihenanalyse miteinander zu verbinden. Weiterhin werden in [20] ebenfalls Untersuchungen zu RNN durchgeführt. Da dabei jedoch nicht die Eingangsgrößen rückgekoppelt werden, sondern die Zustände der Neuronen in den verdeckten Schichten, ist eine Analyse dieser Art neuronaler Netze schwerer zu interpretieren, weil die Neuronenzustände nicht mehr zwingend Abbildungen physikalischer Größen sind. Dies könnte Aufgabe zukünftiger Arbeiten sein.

In der Arbeit von [20] werden unterschiedliche Netze zur Spurwechselprediktion entwickelt. Hierzu werden verschiedene Netzwerktypen, unterschiedliche Netzwerktopologien und der Einfluss vieler Hyperparameter empirisch untersucht. Mit dem besten in dieser Arbeit entwickelten Netz können Spurwechsel im Testdatensatz der Simulationsdaten durchschnittlich nach 28 % der Dauer des Labels erkannt werden, was einer Prädiktionszeit von ca. 1,4 s vor dem eigentlichen Spurwechsel entspricht.

Die empirische Art der Bestimmung der Hyperparameter entspricht der typischen Vorgehensweise bei der Entwicklung neuronaler Netze. Das Ziel des vorliegenden Beitrags soll es sein, analytische Methoden zur Optimierung neuronaler Netze aufzuzeigen.

5 Analytische Verfahren zur Optimierung der Netzwerktopologie

5.1 Untersuchungen

Die Hyperparameter eines neuronalen Netzes können in zwei Gruppen eingeteilt werden: die Lernparameter und die topologischen Parameter. Während die Lernparameter nur während der Trainingsphase des Netzes einen direkten Einfluss auf das Ergebnis der Berechnung des Netzwerkes haben, wirken die topologischen Parameter sowohl während der Trainingsphase als auch während der Ausführungsphase direkt auf den Netzausgang ein. Die

Lernparameter beeinflussen den Optimierungsprozess der Gewichte und Bias und tragen durch das Identifizieren der finalen Werte der Netzwerkparameter nur indirekt zum Netzwerkausgang während der Ausführungsphase bei. Die topologischen Parameter hingegen beeinflussen durch die Vorgabe der Gestalt des Modells des neuronalen Netzes die Berechnung des Netzwerkausgangs sowohl während der Trainingsphase als auch während der Ausführungsphase.

Ziel dieses Beitrags ist es, die in [20] empirisch ermittelten topologischen Parameter analytisch zu erklären beziehungsweise diese analytisch zu optimieren. Vor dem Hintergrund des beabsichtigten Einsatzfalls für das automatisierte Fahren bedeutet eine Optimierung nicht zwangsläufig eine Steigerung der Erkennungsrate sondern auch eine Reduktion der topologischen Parameter, da sich dies positiv auf die Rechenzeit und den Speicherbedarf auswirkt. Kennwerte, die zur Bewertung verschiedener Netze herangezogen werden können, sind somit einerseits leistungsspezifische Größen wie die Zeitdifferenz zwischen erkanntem Spurwechsel und Label des Spurwechsels und andererseits architektonische Kenngrößen wie die Anzahl der Netzwerkparameter. An einem in [20] trainierten Netz soll untersucht werden, ob dieses Netz mit 150 Eingangneuronen, 75 Neuronen in einer verdeckten Schicht und drei Ausgangsneuronen eine optimale Topologie aufweist. Da auf Grund des Einsatzfalles die Art des Ausganges mit drei verschiedenen möglichen Ausgängen vorgegeben ist, soll untersucht werden, wie sich die Neuronen der Eingangsschicht und der verdeckten Sicht auf den Netzausgang auswirken und ob gegebenenfalls die Anzahl der Neuronen in diesen Schichten geändert werden sollte.

5.2 Auswertung

Entsprechend des Aufbaus und der Ausgabe des trainierten neuronalen Netzes kann die Verteilung der Eingangsdaten hinsichtlich des Netzausganges nach zwei verschiedenen Kriterien untersucht werden. Einerseits können die Verteilungen der Eingangsdaten (150 Merkmale in 137.520 Samples) entsprechend der Wahrscheinlichkeit der Prädiktion der drei verschiedenen Ausgänge gruppiert werden, beispielsweise schrittweise je 10 % Erkennungswahrscheinlichkeit. Andererseits können die Verteilungen der Eingangsdaten auch entsprechend der Zeitschritte der Historie der Daten separiert werden, sodass die Merkmale aus Tabelle 1 je 0,1 s Zeitschritt bezüglich des Einflusses auf den Netzausgang untersucht werden können.

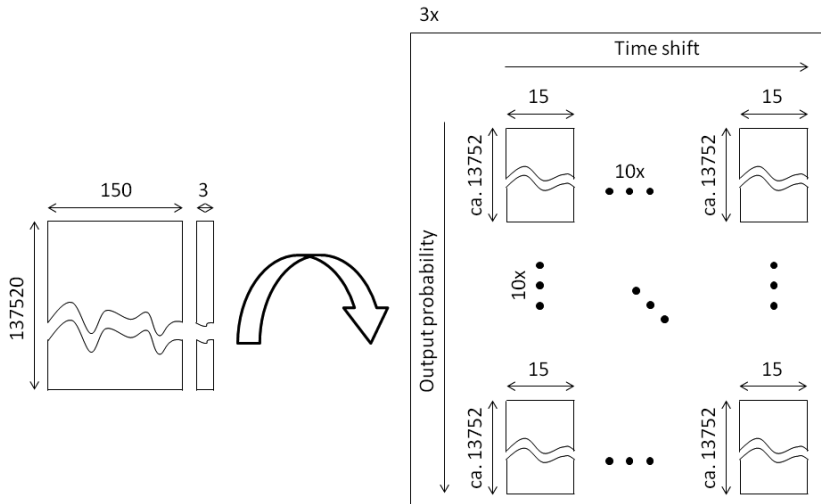


Bild 3: Darstellung der Aufteilung der Eingangsdaten entsprechend der Zeitschritte der Historie und der Wahrscheinlichkeitsklassen der Erkennungswahrscheinlichkeiten des Netzausganges

Die Aufteilung der Daten gemäß dieser Beschreibung ist in Abbildung 3 visualisiert.

Exemplarisch sollen die Verteilungen des Merkmals laterale Geschwindigkeit bezüglich der beiden Unterscheidungsmethoden betrachtet werden. Den Abbildungen 4 und 5 ist zu entnehmen, wie sich die Verteilungskurven der relativen Häufigkeiten der Ausgangsklassen kein Spurwechsel, Spurwechsel links und Spurwechsel rechts abhängig von der Wahrscheinlichkeit mit der sie ausgegeben werden, verändern.

Zu erkennen ist, dass die drei Wahrscheinlichkeitsverteilungen bei einer Prädiktionswahrscheinlichkeit von 90 - 100 % gut voneinander separierbar sind. Aus Grafik 4 kann gefolgert werden, dass sich bei hohen Erkennungswahrscheinlichkeiten eine bestimmte laterale Geschwindigkeit eindeutig einem Spurwechselzustand zuordnen lässt. Bild 5 verdeutlicht dagegen, dass bei einer geringen Prädiktionswahrscheinlichkeit von 0 - 10 %, ein bestimmter Wert für die laterale Geschwindigkeit nicht eindeutig einem Netzausgang zugeordnet werden kann, da die Verteilungskurven für Spurwechsel links und Spurwechsel rechts fast exakt übereinander liegen. Der Wertebereich des

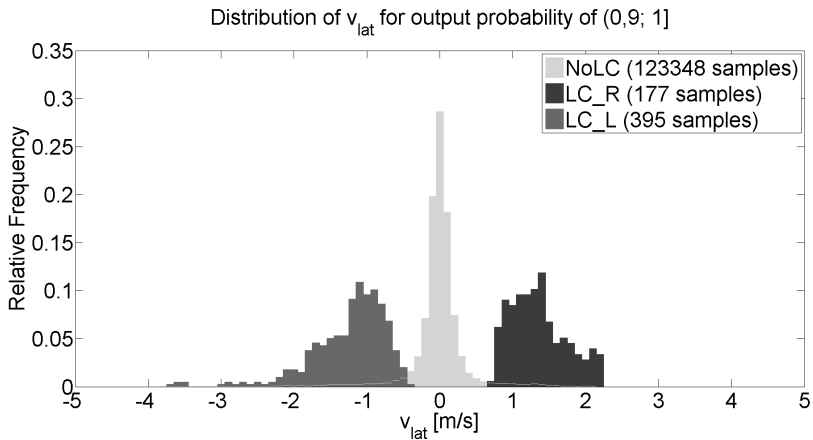


Bild 4: Verteilung des Merkmals laterale Geschwindigkeit bei hoher Erkennungswahrscheinlichkeit $P > 0,9$

Merkmals laterale Geschwindigkeit ändert sich demzufolge in Abhängigkeit der Erkennungswahrscheinlichkeit des jeweiligen Netzausganges.

Abbildung 6 stellt dar, dass die Betrachtung des Wertes der lateralen Geschwindigkeit 0,9 s vor dem aktuellen Messwert nur eingeschränkt weitere Informationen über die Spurwechselabsicht enthält.

Aus der Visualisierung geht hervor, dass sich die Wahrscheinlichkeitsverteilungen für alle drei möglichen Netzausgänge sehr ähneln. Für eine bestimmte laterale Geschwindigkeit zu diesem Messzeitpunkt werden alle drei Netzausgänge somit etwa zu 33 % erkannt.

Werden den Histogrammen (Normal-)Verteilungskurven angepasst, so können über die Angabe statistischer Merkmale wie des Mittelwertes und der Standardabweichung quantitative Aussagen zum Einfluss der einzelnen Merkmale abgeleitet werden.

Ähnliche Überlegungen zur Bestimmung der Relevanz einzelner Neuronen zur Prädiktion eines Netzausganges, jedoch für Neuronen der verdeckten Schicht, sollen im Folgenden betrachtet werden. Der Untersuchung, wie stark der Ausgang jedes Neurons der verdeckten Schicht auf einen bestimmten Netzausgang wirkt, liegt die Idee zu Grunde, Neuronen der verdeckten Schicht zu entfernen, falls gezeigt werden kann, dass diese keinen wesentlichen Beitrag zur Erkennung aller Zustände des Netzausgangs liefern (alle

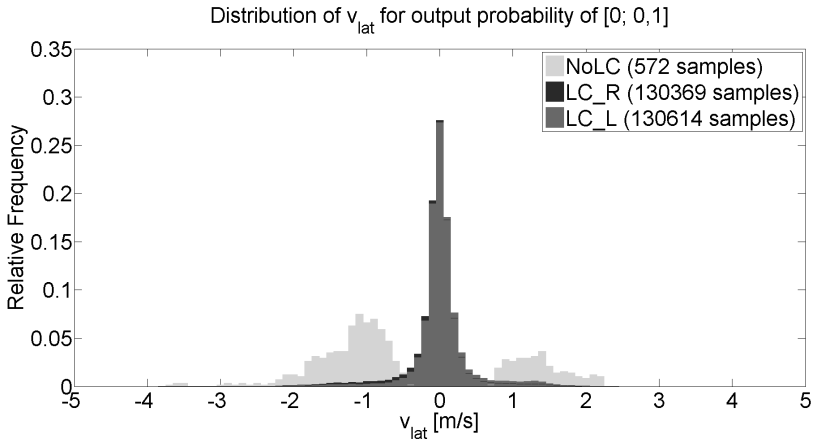


Bild 5: Verteilung des Merkmals laterale Geschwindigkeit bei geringer Erkennungswahrscheinlichkeit $P \leq 0,1$

drei möglichen Netzwerkausgänge für alle Wahrscheinlichkeitsintervalle). Auf diese Weise können systematisch und schichtweise rückwärts gerichtet (vom Netzausgang beginnend) Neuronen aus vorhergehenden Sichten entfernt werden, falls sich diese als für die weitere Funktionsweise des Netzes unbedeutend herausgestellt haben. Hier werden nun dazu die Ausgänge der Neuronen x_i der verdeckten Schicht $l - 1$ pro Ausgangsneuron j auf die Summe der Neuronenausgänge für das betreffende Ausgangsneuron normiert. Dieser Wert kann als Beitrag zur Aktivierung der nächsten Schicht a_{ji}^l interpretiert werden (9):

$$a_{ji}^l = w_{ji}^l \frac{|o_i^{l-1}|}{\sum_i |o_i^{l-1}|} \quad (9)$$

Da die Gewichte w_{ji} negativ sein können, sollen die einzelnen Beiträge zur Aktivierung der Neuronen der Ausgangsschicht hinsichtlich ihres Vorzeichens unterschieden werden. Die für die weitere Analyse wichtigere Information stellt jedoch der Betrag der Werte dar, da dieser darüber entscheidet, ob das jeweilige Neuron zur Sicherstellung der Funktion des Netzes notwendig ist. Für das in dem vorherigen Abschnitt betrachtete neuronale Netz aus [20] ist dies exemplarisch in Abbildung 7 für eine jeweilige Ausgabewahrscheinlichkeit von 80 - 90 % dargestellt.

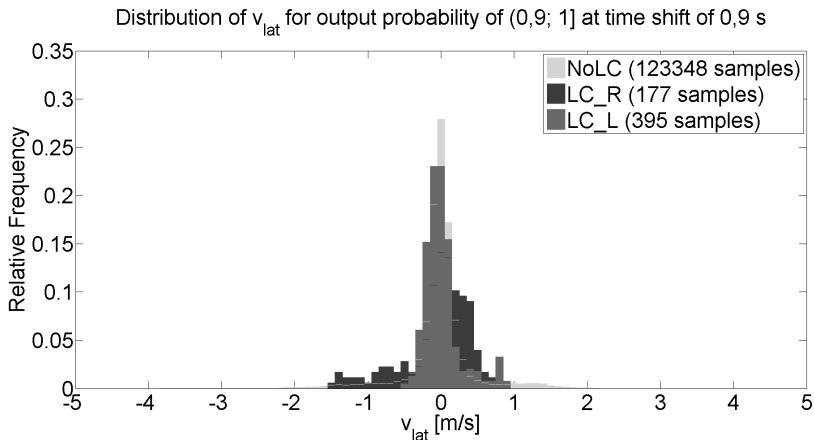


Bild 6: Verteilung des Merkmals laterale Geschwindigkeit 0,9 s vor dem aktuellen Messwert bei hoher Erkennungswahrscheinlichkeit $P > 0,9$

Tabelle 2: Vergleich der Fehlerrate Error Prediction Rate beider Netze auf dem Testdatensatz

Neuronales Netz	Error Prediction Rate [%]
Verdeckte Schicht mit 75 Neuronen	3,2701
Verdeckte Schicht mit 74 Neuronen	3,4024

Bei der in Abbildung 7 gewählten Form der Darstellung entspricht der Betrag des berechneten Wertes dem Radius der Kreise im Bild. Je kleiner der Punkt, der zu einem bestimmten Neuron gehört, ist, desto kleiner ist auch sein Beitrag zu Aktivierung des entsprechenden Ausgangsneurons. Der gewichtete Ausgang des vierten Neurons in der verdeckten Schicht ist nicht nur für die Ausgangswahrscheinlichkeiten von 80 - 90 % für alle drei Ausgaben äußerst gering, sondern auch für die übrigen neun Wahrscheinlichkeitsbereiche ($\min = 5,16 \cdot 10^{-5}$; $\max = 7,17 \cdot 10^{-3}$). Daher soll ein Netz aufgestellt werden, welches statt bisher 75 Neuronen in der verdeckten Schicht nur noch 74 Knoten besitzt. Tabelle 2 gibt die Error Prediction Rate als Fehlermaß für den Testdatensatz der beiden Netze an.

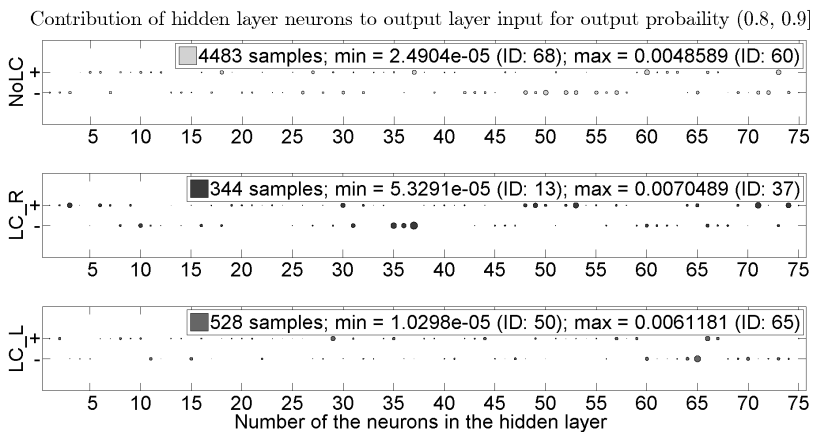


Bild 7: Beiträge der Neuronen der verdeckten Schicht zum Eingang der Ausgangsneuronen

Der absolute Unterschied in der Fehlerrate beträgt zwar nur 0,13 % doch ist die Fehlerrate durch die Reduzierung der Anzahl der Neuronen in der verdeckten Schicht relativ um 3,89 % gestiegen. Wird Abbildung 8 mit Bild 7 verglichen, so ist zu erkennen, dass sich durch das Trainieren des neuen Netzes mit nur 74 Neuronen in der verdeckten Schicht ein völlig anderes Netzwerk ergeben hat. Die beiden Grafiken zeigen beispielsweise, dass die Neuronen mit großer Aktivierung in der verdeckten Schicht jeweils unterschiedliche Positionen innerhalb dieser Schicht einnehmen.

Ob dies ganzheitlich betrachtet eine Optimierung entsprechend der vormals genannten Kriterien des neuronalen Netzes darstellt, sollte in weiteren Untersuchungen ausgewertet werden. Eine Alternative zum gewählten Vorgehen, bei dem das Netz komplett neu trainiert wurde, stellt die Nutzung der bereits vom ersten Netz gelernten Gewichte und Bias zur Initialisierung des neuen Netzwerkmodells dar. Nach geeigneter Anpassung der Netzwerkparameter (Weglassen der überflüssigen Netzparameter des alten Netzwerkmodells) kann das neue Netz anschließend für eine bestimmte Anzahl an Epochen weiter trainiert werden.

Contribution of hidden layer neurons to output layer input for output probability (0.8, 0.9]

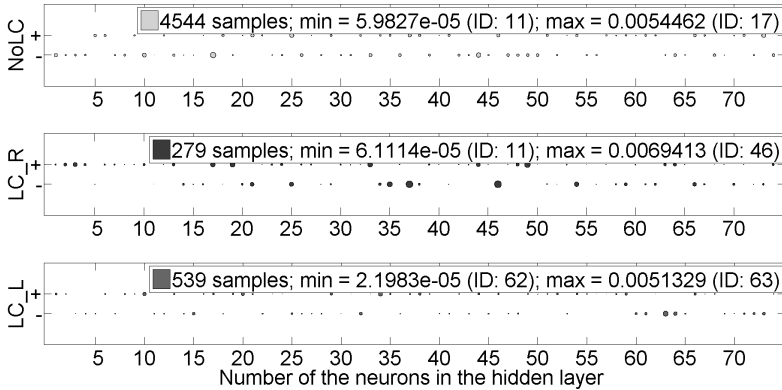


Bild 8: Beiträge der Neuronen der verdeckten Schicht zum Eingang der Ausgangsneuronen für ein Netz mit 74 Neuronen in der verdeckten Schicht

6 Zusammenfassung und Ausblick

Die vorliegende Arbeit untersucht wie ein in [20] empirisch ermitteltes neuronales Netz zur Spurwechselprädiktion, systematisch hinsichtlich seiner topologischen Struktur untersucht werden kann. Hierzu werden Analysen bezüglich des Zusammenhangs zwischen dem Ausgang des Netzes und dem Netzeingang beziehungsweise dem Ausgang verdeckter Neuronenschichten angestellt. Es wird gezeigt, wie sich die Struktur eines neuronalen Netzes vereinfachen lässt, falls es zu Beginn mit zu vielen überschüssigen Netzwerkparametern initialisiert wurde. Die Ergebnisse bestätigen, dass in dem gegebenen Netz Potential zur Reduzierung der Anzahl der verwendeten Neuronen vorhanden ist. In zukünftigen Arbeiten sind die Einflüsse der Netzwerktopologie auf den Netzausgang weiter zu quantifizieren, sodass darauf aufbauend detailliertere Anpassungen der Netzstruktur vorgenommen werden können, ohne die Erkennungsrate des Netzes zu senken.

Danksagung

Stefan Meuresch hat im Rahmen seiner Masterarbeit die numerischen Ergebnisse der Netzsimulationen erzeugt, hierfür bedanken sich die Autoren.

Literatur

- [1] Statistisches Bundesamt. „Verkehr Verkehrsunfälle, Fachserie 8 Reihe 7“. Wiesbaden. 2016.
- [2] Verband der Automobilindustrie e.V.. „Von Fahrerassistenzsystemen zum automatisierten Fahren“. Berlin. 2015.
- [3] Society of Automotive Engineers. „Automated Driving Levels of Driving Automation are Defined in New SAE International Standard J3016“. Warrendale. 2014.
- [4] Y.L. Murphey, R. Milton, L. Kiliaris. „Driver’s Style Classification Using Jerk Analysis“. Computational Intelligence in Vehicles and Vehicular Systems, 2009. IEEE Workshop on, Nashville, TN, S. 23-28. 2009.
- [5] T. Hagiwara, R. Sakakima, T. Hashimoto, T. Kawai. „Effect of Distraction on Driving Performance using Touch Screen while Driving on Test Track“. Intelligent Vehicles Symposium, 2013 IEEE, Gold Coast, QLD, S. 1149-1154. 2013.
- [6] D.L. Strayer, F.A. Drews, D.J. Crouch. „A Comparison of the Cell Phone Driver and the Drunk Driver“. Human factors: The journal of the human factors and ergonomics society, 48, 2, 381-391, Sage Publications. 2006.
- [7] I. Dagli, G. Breuel, H. Schittenhelm, A. Schanz. „Cutting-in vehicle recognition for ACC systems - towards feasible situation analysis methodologies“. Intelligent Vehicles Symposium, 2004 IEEE, S. 925-930. 2004.
- [8] H.M. Mandalia, D.D. Salvucci. „Using Support Vector Machines for Lane-Change Detection“. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 49; S. 1965–1969. 2005.
- [9] Ü. Dogan, J. Edelbrunner, I. Iossifidis. „Autonomous Driving: A Comparison of Machine Learning Techniques by Means of the Prediction of Lane Change Behavior“. Robotics and Biomimetics, 2011 IEEE Intern. Conference on, Karon Beach, Phuket, S. 1837-1843. 2011.
- [10] B. Morris, A. Doshi, M. Trivedi. „Lane Change Intent Prediction for Driver Assistance: On-Road Design and Evaluation“. Intelligent Vehicles Symposium, 2011 IEEE, Baden-Baden, S. 895-901. 2011.

- [11] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, W. Rosenstiel. „Object-oriented Bayesian networks for detection of lane change maneuvers“. Intelligent Vehicles Symposium, 2011 IEEE, Baden-Baden, S. 673-678. 2011.
- [12] P. Kumar, M. Perrollaz, S. Lefevre, C. Laugier. „Learning-based approach for online lane change intention prediction“. Intelligent Vehicles Symposium, 2013 IEEE, Gold Coast, QLD, S. 797-802. 2013.
- [13] C. Wang, J. Qin, R. Zhang, Y. Zhang. „Application of Support Vector Machine in Lane Change Recognition“. Journal Of Theoretical & Applied Information Technology, 48, 3, S. 1449-1457. 2013.
- [14] J. Schlechtriemen, A. Wedel, J. Hillenbrand, G. Breuel, K.D. Kuhnert. „A lane change detection approach using feature ranking with maximized predictive power“. 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, S. 108-114. 2014.
- [15] J. Firl. „Probabilistic Maneuver Recognition in Traffic Scenarios“. Dissertation. Karlsruher Institut für Technologie. Fakultät für Maschinenbau. Institut für Mess- und Regelungstechnik. 2014.
- [16] T. Wheeler. „Probabilistic Driving Models and Lane Change Prediction“. Stanford University. 2014.
- [17] G.D. Rey, K.F. Wender. „Neuronale Netze: Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung“. Hans Huber Verlag. Bern. 2011.
- [18] R. Rojas. „Neuronal Networks a Systematic Introduction“. Springer Verlag. Berlin. 1996.
- [19] J. Exner. „Untersuchung von Gauß-Prozessen zur Fahrmanöverdetektion auf Autobahnen“. Masterarbeit. TU Dortmund. Lehrstuhl für Regelungssystemtechnik. 2015.
- [20] S. Meuresch. „Künstliche neuronale Netze für die Fahrsituationserkennung in automatischen Fahrfunktionen“. Masterarbeit. RWTH Aachen. Fachbereich Maschinenbau. Institut für Kraftfahrzeuge. Arbeitsstand 22. September 2016.
- [21] A Agarwal et al.. „An Introduction to Computational Networks and the Computational Network Toolkit“. Microsoft Technical Report MSR-TR-2014-112, Redmond 2014.

Data Preprocessing: A New Algorithm for Univariate Imputation Designed Specifically for Industrial Needs

Sowmya Chandrasekaran, Martin Zaefferer, Steffen Moritz,
Jörg Stork, Martina Friese, Andreas Fischbach,
Thomas Bartz-Beielstein

SPOTSeven Lab, TH Köln

Steinmüllerallee 1, 51643 Gummersbach

E-Mail: {sowmya.chandrasekaran, martin.zaefferer, steffen.moritz, joerg.stork,
martina.friese, andreas.fischbach, thomas.bartz-beielstein}@th-koeln.de

Abstract

Data pre-processing is a key research topic in data mining because it plays a crucial role in improving the accuracy of any data mining algorithm. In most real world cases, a significant amount of the recorded data is found missing due to most diverse errors. This loss of data is nearly always unavoidable. Recovery of missing data plays a vital role in avoiding inaccurate data mining decisions. Most multivariate imputation methods are not compatible to univariate datasets and the traditional univariate imputation techniques become highly biased as the missing data gap increases. With the current technological advancements abundant data is being captured every second. Hence, we intend to develop a new algorithm that enables maximum utilization of the available big datasets for imputation. In this paper, we present a Seasonal and Trend decomposition using Loess (STL) based Seasonal Moving Window Algorithm, which is capable of handling patterns with trend as well as cyclic characteristics. We show that the algorithm is highly suitable for pre-processing of large datasets.

1 Introduction

Data pre-processing involves removal of noise and outliers from a dataset, handling of missing values, data redundancy and data inconsistency. One of

the most challenging task among them is to impute the missing values with entries that reasonably complete the datasets. The loss of data may be due to sensor errors, transmission errors, errors of the operator and other errors. Recovery of these missing values heavily affects the performance of the data mining models. The accuracy of forecasting, classification, estimation, and pattern detection of any data mining algorithm depends significantly on the accuracy of data used in modeling. Thus, inaccurate training and testing data may introduce bias into the models and provide misleading conclusions [14]. In reality, datasets are commonly univariate. Moreover, multivariate datasets may lack correlation. The need for univariate time series analysis is prevalent in many fields, for example, online data monitoring and pattern detection in intensive care units [1], forecasting in hydrology and environmental management fields [2], functional magnetic resonance imaging statistical analysis [3], forecasting intra day arrivals at a call center [4], forecasting electricity spot-prices [5], forecasting macroeconomic time series [6].

With univariate time series data, the complexity of replacing these missing value increases as no correlated variables are available. Almost all of the well known standard techniques fail to handle univariate time series data as their scheme is based on the inter-attribute correlations in estimating the values for the missing data. Also, the existing multivariate algorithms either cannot be applied or perform poorly. Furthermore, some traditional imputation techniques perform well with trend datasets, while some techniques perform only well if the dataset is seasonal. There exists no single univariate imputation technique that is suitable for all types of data patterns [7]. This is owing to the reason that most of the existing algorithms are designed extensively to handle either seasonality or trend and not both.

The major motivation for this work is the GECCO Industrial Challenge 2015. The task was to recover the missing data in heating systems (<http://www.spotseven.de/gecco/gecco-challenge/gecco-challenge-2015/>). The data contains 606,837 observations of four parameters sampled every minute from real industrial heating systems. The most challenging aspect of this challenge is to impute the missing data for a large interval with the all data missing. This is a commonly occurring real world scenario, when there is some data transmission failure, file over writing or data saving issue requiring univariate imputation. Also, such scenarios lead to large intervals of missing data for which the standard imputation techniques perform poorly.

The remainder of this paper is organized as follows. Section 2 focuses on the proposed imputation technique. Section 3 illustrates the experimental study and performance comparisons among various imputation schemes. Section 4 presents our concluding remarks.

2 Proposed Algorithm

The new algorithm entitled *Seasonal Moving Window Algorithm* (SMWA) is proposed mainly for large intervals of missing data, especially seasonal and cyclic data. The key aspect of our algorithm is that strong seasonality exists in almost all practical applications. Moreover, this seasonal behavior has to be considered as cyclic. There is no guarantee that the behavior of a system at a specific time is identical on two different days. Although it is very likely that the system behavior will be similar, regardless of the exact time. SMWA utilizes Seasonal and Trend decomposition using Loess (STL) [10] to decompose the data. Our proposed approach differs from other existing STL based technique [8] in how we handle imputation after performing STL decomposition: The decomposed trend component is linearly interpolated. The seasonal and remainder component is fitted with best pattern identified from the past available data. Then, the imputed decomposed data is recomposed to form the complete dataset.

SMWA initially identifies the missing interval as *missing data*. Then it selects a finite set of data before the missing interval as *head* and in a similar fashion chooses a finite set of data after the missing interval as *tail*. The combination of *head*, *missing data*, and *tail* forms the *window* as shown in Figure 1. This *window* then slides through the past data and the best matching window with the minimum root mean square error (RMSE) is imputed in the *missing data*.

The data preparation for SMWA imputation is explained in Algorithm 1. A univariate time series ts is given as the input dataset. The input time series is first validated for the presence of missing values. Then, the indexes of the missing data are identified.

To implement the algorithm, we first perform seasonal decomposition with `stl` [10]. As we require a complete dataset for `stl`, linear interpolation is performed as described in [11]. After the data is decomposed into seasonal, trend, and irregular components, the trend component is separated. Then,

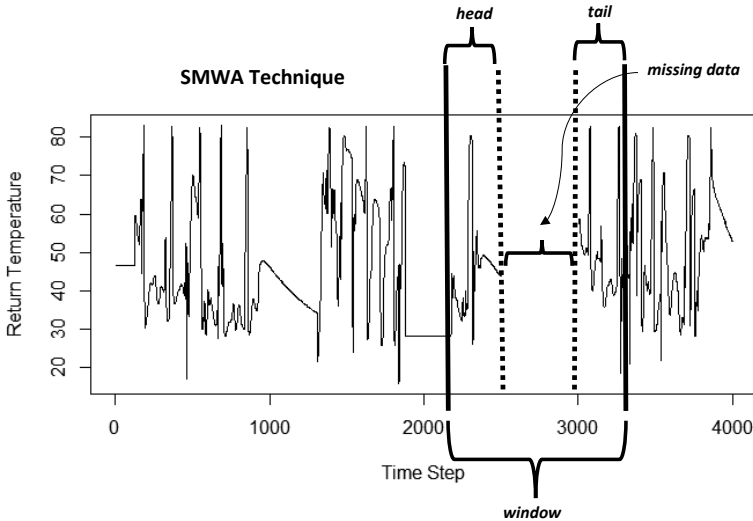


Figure 1: Pictorial representation of the SMWA Technique

SMWA is applied to the remaining component (i.e., seasonal and irregular) denoted as *SMWAInput*. In the separated trend component, missing values are filled in using linear interpolation as in [11]. Since the trend represents the long term rise or drop in data, simple interpolation is sufficient to fit the missing data in the trend component.

The SMWA imputation is explained in Algorithm 2. It uses *SMWAInput* obtained from Algorithm 1. Based on the length of data and the missing values a value for l ($l=head=tail$), which represents the common size of the dataset for both *head* and *tail*, has to be provided. The minimum length of the number of missing data samples g , for which the imputation has to be done, has to be provided. For smaller missing intervals, i.e., less than g missing samples, linear interpolation is performed. The user is also free to choose the maximum length of past window w to be considered for the evaluation. The default suggestions based on preliminary experiments are past window size w of $\frac{n}{3}$ and l of $\frac{w}{12}$ for smaller datasets, where n represents the length of the dataset.

Let $\text{freq}(ts)$ denote the frequency of the dataset. For larger datasets, e.g, $n > 100,000$, default suggestions are the past window w of size $\frac{n}{30}$ and l of $\frac{n}{(60 \times \text{freq}(ts))}$ for minutely data, for hourly data l of $\frac{n}{(60 \times \text{freq}(ts))}$, and for other

Algorithm 1.: Data preparation for the Seasonal Moving Window Algorithm (SMWA)

Voraussetzung: univariate Time Series ts

- 1: Validate the input data
- 2: Determine the indexes of missing samples: $index$
- 3: Perform linear interpolation on ts
- 4: Decompose interpolated ts into $Seas$, $Trend$, $Irreg$ components with stl
- 5: Separate the $Trend$, fill in NA in $index$, perform linear interpolation
- 6: Add $Seas$, $Irreg$ as $SMWAInput$

Abschlussbedingung: $SMWAInput$, $Trend$,

Algorithm 2.: Seasonal Moving Window Algorithm (SMWA)

Voraussetzung:

univariate Time Series ts

l ▷ size of dataset for $head$ and $tail$

g ▷ minimum length of missing gap to be imputed

w ▷ maximum length of past window

$option$ ▷ accepts string $head$, $tail$, $both$

$SMWAInput$ and $Trend$ from Algorithm 1

- 1: Calculate n as length of ts
- 2: **for** i in $1:n$ **do**
- 3: Identify the $indices$ with missing intervals $\geq g$ in ts
- 4: **for** $eachindex$ in $indices$ **do**
- 5: Identify the actual $missingdata$ in ts
- 6: Formulate $window$ as $head + missingdata + tail$ ▷ as per specifications in $option$
- 7: **for** j in $1:w$ **do**
- 8: **Try**{ ▷ try-catch to ensure if past window of specified length ' w ' exist before $missingdata$
- 9: Slide the $window$ in the past by j in ts and calculate the RMSE for $head$ and $tail$ }
- 10: **Catch**{
- 11: Notify error }
- 12: Find the best fitting past window with least RMSE for $head$ and $tail$
- 13: **return** The best fitting window

13: $SMWAImputed \leftarrow$ Impute the values from the best fitting window in the $missingdata$ in ts

Abschlussbedingung: $SMWAResult \leftarrow Trend + SMWAImputed$ ▷ Final SMWA imputed time series

frequencies l of $\frac{w}{10}$. Considering the computational time, we recommend this algorithm for fairly large g . This algorithm can be computed with either *head* only or *tail* only, or *both*. The performance of the method depends on the choice of tuning parameters l , g and w . The value of the input parameters depends on the length of the dataset and the percentage of the missing values and hence it might vary for each dataset. The window is formed with *head*, *missing data*, and *tail* for each of the missing intervals greater than g . Then we slide this window through past data, but no earlier than w steps before the window. The best matching past window with the smallest *root mean squared error* (RMSE) is identified and the values of this best match are imputed into the gap. Finally, the trend component is added back into the imputed dataset.

3 Experimental Study

In all our datasets, as the probability of missing data does not depend upon the observed or the unobserved data these are classified as *missing completely at random* (MCAR) [9]. As this algorithm is proposed mainly for large intervals of missing data, we examine this feature by removing relatively large data intervals. As the performance of the algorithms may depend on the position of missing values, the missing intervals were chosen randomly each time based on 30 different random seeds. The performance of the algorithm is evaluated with various test scenarios. For each test scenario the following steps are performed:

1. Load a complete time series *tsComplete*
2. Randomly remove values in *tsComplete* as per each scenario requirements and obtain *tswithNAs*
3. Apply an imputation algorithm to *tswithNAs* to get *tsImputed*
4. Compare *tsComplete* and *tsImputed* by using a suitable accuracy or error measure

3.1 Comparison Framework

To analyze the efficiency of the SMWA algorithm, its performance is compared with several state of the art imputation techniques which were

implemented in the statistical programming language R. The RMSE was chosen as a performance measure, i.e.,

$$RMSE(z, z_{\text{imp}}) = \sqrt{\frac{\sum_{t=1}^n (z - z_{\text{imp}})^2}{n}} \quad (1)$$

where z_{imp} is the imputed value and z denotes the actual value of the time series. Mainly, methods from `imputeTS` [12], `zoo` [11], and `forecast` [8] packages in R are used for experiments as the described below:

- **Spline interpolation:** This method from the `imputeTS` package uses `na.interpolation` to implement the spline interpolation of missing values.
- **Seasonal decomposition:** This method is also from the `imputeTS` package. It initially separates the seasonal component from the time series, then performs imputation on the trend and irregular components and finally adds the seasonal component again. The method used is `na.seadec`. The algorithm internally uses mean imputation for nonseasonal series.
- **Seasonal split:** This is the third method from the `imputeTS` package. It splits the times series into seasons and then performs imputation separately for each of the season. The algorithm used is `na.seasplit`. The algorithm internally uses mean imputation for non seasonal series.
- **LOCF:** LOCF stands for last observation carried forward. It is a method from the `zoo` package. It replaces each missing value with the most recent non missing value prior to it. The algorithm used is `na.locf`.
- **Mean Imputation:** This method is from the `zoo` package. It fills the missing values with mean value of a time series using `na.aggregate`.
- **Linear interpolation:** It is a method from the `zoo` package. It replaces the missing values with interpolated values using `na.approx`.
- **Structural time series model:** It is a method from the `zoo` package. It fills missing values using seasonal Kalman filter using `na.StructTS`.
- **STL based interpolation:** It is a method from the `forecast` package, which uses linear interpolation for non-seasonal series and a periodic STL decomposition with seasonal series. The method used is `na.interp`.

- **Kalman smoothing:** This method is also from the `imputeTS` package. It performs Kalman smoothing using the state space representation of an ARIMA model for imputation. The method used is `na.kalman` with `auto.arima` model.

3.2 Case I: Large real-world data set

The proposed SMWA technique was evaluated for the *Return Temperature* dataset from GECCO Industrial Challenge 2015 which consists of 606,837 observations. The challenge provides separate missing and complete datasets. The algorithm is implemented for a minimum gap of $g = 200$. For the rest of the missing data linear interpolation is performed. The preceding 400 values form the *head* and after the missing gap and succeeding 400 values form the *tail* ($l = 400$). This *window*, formed with *head + missing data + tail*, is moved through the previous $w = 50,000$ values and the window with the best match according to RMSE is chosen.

The results for various algorithms tested are described in Table 1. A RMSE of 5.56 was achieved by SMWA when compared to the original test dataset. Although little improvement in RMSE was obtained for the proposed scheme when compared with linear interpolation, a closer look into the imputed intervals reveals the improvement made.

The best performing algorithms are visualized for some large missing intervals in Figures 2 and 3. They show the improvement achieved with SMWA imputation. From the plots it can be seen that SMWA outperforms other competing algorithms for large intervals of missing data. It also reveals the ability of SMWA in determining the best similar pattern as in the original underlying test dataset. By visual analysis, it can be seen that SMWA is often (but not always) able to reproduce the more complex patterns observed in the data. The competing methods mostly fail to do so. Still, such an imputed pattern will often be shifted forward or backward. Thus, the evaluation by RMSE may fail to give proper credit to the SMWA. In practice, the pattern is more desirable than a good value of RMSE, we therefore propose to use a more adequate error measure. For instance, an alignment-based error measure like the *Dynamic Time Warping distance* might be more adequate in these situations [15].

Table 1 also shows that conventional imputation schemes like mean or loef imputation require less time for computation, but their RMSE is very high. The Kalman smoothing achieved the second best algorithm. But

Table 1: Comparison of RMSE and computation time. Results on the return temperature (GECCO challenge) dataset for various imputation techniques are shown. Smaller values are better. The structural time series model algorithm failed to complete the imputation process. Best values are shown in boldface.

Imputation Method	RMSE	Computation time (s)
Mean Imputation	15.59	0.33
Seasonal split	13.83	16.69
Seasonal decomposition	13.32	576.98
Spline interpolation	13.25	561.27
LOCF	8.81	11.28
STL based interpolation	6.74	72.02
Linear interpolation	5.60	11.55
SMWA imputation	5.56	143.89
Kalman smoothing	5.58	9134.15
Structural time series model	NA	NA

Kalman smoothing is computationally demanding (a factor of 60 compared to SMWA imputation). The structural time series model algorithm failed to complete the imputation process for the given data for more than 12 hours and hence the scheme was suspended and the result using this scheme is not discussed.

3.3 Case II: Selected smaller data sets

The algorithm is tested for its adaptability to various kinds of data patterns. Three different datasets that exhibit specific data patterns are considered. The SP dataset [13] (with trend but no seasonality), the Beer-sales dataset [13] (no trend but with seasonality), and the Air Passengers dataset [13] (with trend and seasonality).

Since the size of these datasets is very small, 10% of the datasets is removed in a single interval. This missing interval is determined randomly for 30 repeated experiments. The SMWA parameters l and w for three datasets are chosen as per default suggestions for smaller datasets.

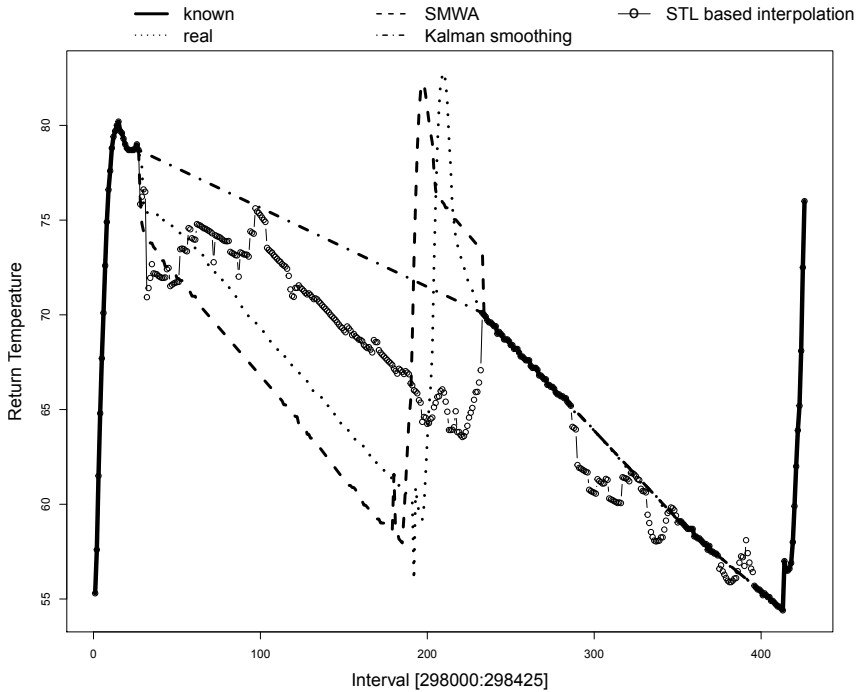


Figure 2: Comparison of the best performing algorithms for the Return Temperature dataset. The RMSE[298000:298425] for SMWA is 3.64, Kalman smoothing is 5.17, STL based interpolation is 4.12

3.3.1 Air Passengers dataset

The Air Passengers dataset contains 144 observations with the frequency of time series as 12. Table 2 reveals the mean performance metrics and computational time for the Air Passengers dataset. The box plot of RMSE obtained with 30 repeated experiments for various imputation techniques is shown in Figure 4. In the box plot each algorithm is represented with their abbreviated method names as explained in Section 3.1.

The algorithm Kalman smoothing turns out to be the best followed by STL based interpolation and SMWA. Though SMWA ranks in the third position, it is interesting to note that variance in mean RMSE as shown in the box plot in Figure 4 is less than for the STL based interpolation and

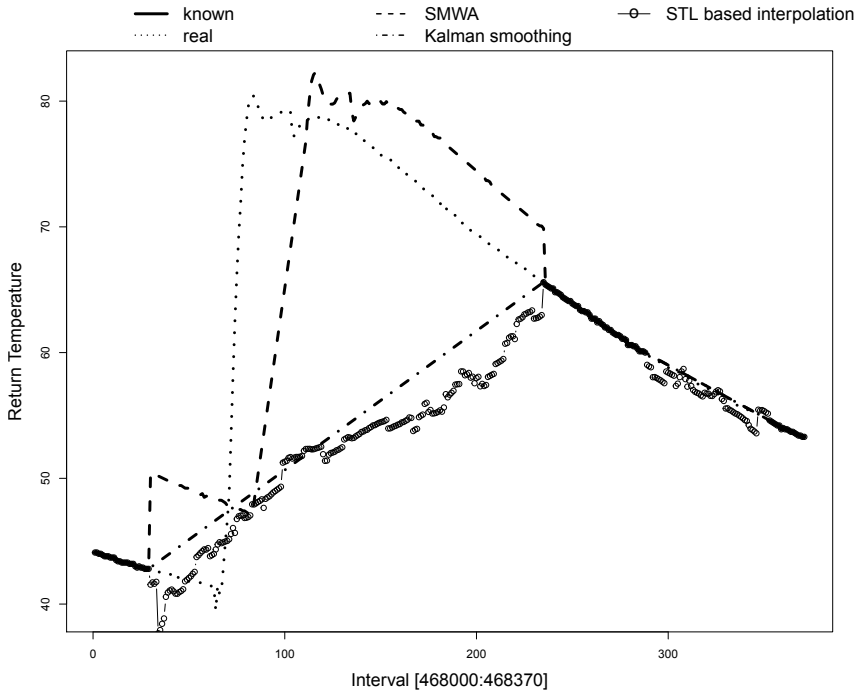


Figure 3: Comparison of the best performing algorithms for the Return Temperature dataset. The RMSE[468000:468370] for SMWA is 7.63, Kalman smoothing is 12.97, STL based interpolation is 13.95

Kalman smoothing. Though the dataset exhibits both trend and seasonality patterns, seasonal decomposition and seasonal split algorithms were able to secure fourth and fifth positions only. The remaining algorithms fail to perform well with relatively continuous missing data.

The Spline interpolation performs badly with very high mean RMSE and also with a very large outlier. The fastest running algorithm is simple the mean imputation and the Kalman smoothing takes the highest computational time.

Table 2: Comparison of RMSE (mean) & Computation time (mean) - Air Passengers dataset for various imputation techniques for 10% of missing data at random locations based on 30 different seeds. Smaller values are better. Best values are shown in boldface.

Imputation Method	RMSE	Computation time (s)
Mean Imputation	52.66	0.002
Seasonal split	40.50	0.01
Seasonal decomposition	39.59	0.02
Spline interpolation	99.18	0.008
LOCF	54.38	0.007
STL based interpolation	13.24	0.03
Linear interpolation	47.09	0.03
SMWA imputation	19.11	0.32
Structural time series model	60.82	0.55
Kalman smoothing	9.75	2.42

3.3.2 Beersales dataset

The Beersales dataset contains 192 observations with a time series of frequency 12. Since the Beersales dataset possesses very high seasonality, it is obvious that seasonal based algorithms like STL based interpolation, seasonal split, and seasonal decomposition would dominate the performance comparisons.

Table 3 shows the mean performance metrics and computational time for the Beersales dataset. The Kalman smoothing and STL based interpolation algorithm both perform the best, followed by seasonal split and seasonal decomposition. Though SMWA ranks in the middle, the performance is very close to the best algorithms. The box plot in Figure 5 shows that all these five algorithms perform better. The Spline interpolation again turns out to be the bad performing one with very high mean RMSE and Kalman smoothing takes the highest computational time.

3.3.3 SP dataset

The SP dataset contains 168 quarterly observations. For the SP dataset, which is the series with just trend and no seasonality, the linear interpolation

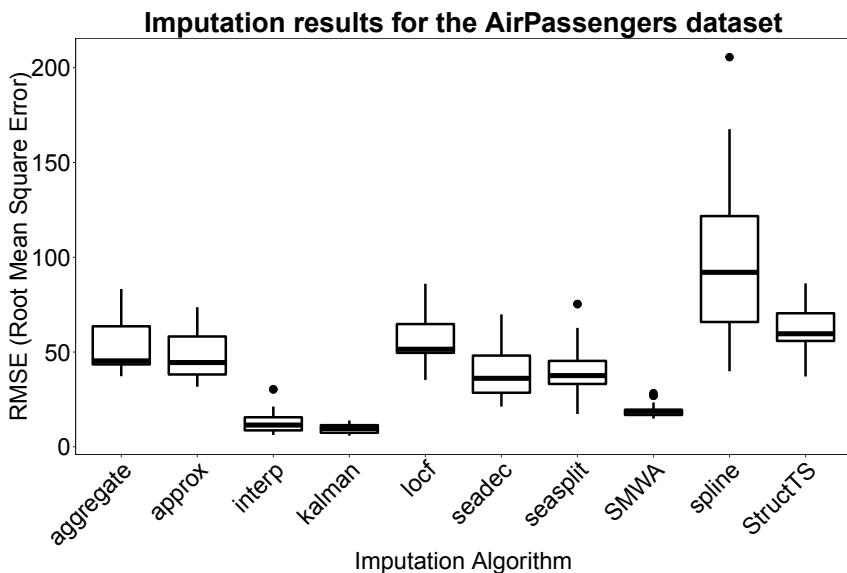


Figure 4: Imputation result for the Air Passengers dataset with 10% of missing data at random locations based on 30 different seeds

algorithm would do a good job. Table 4 lists the mean performance metrics and computational time for the SP dataset. The algorithms Kalman smoothing, STL based interpolation, linear interpolation, SMWA, and structural time series model turn out to be the best performing ones.

It is interesting to note that variance in mean RMSE for the SMWA algorithm is found to be the lowest among other algorithms as shown in the box plot in Figure 6. Since there is no seasonality in SP dataset, the seasonal split and seasonal decomposition algorithms perform just the mean imputation. All other algorithms fail to perform well with relatively continuous missing data. The locf imputation performs badly with very high mean RMSE and Kalman smoothing takes the highest computational time.

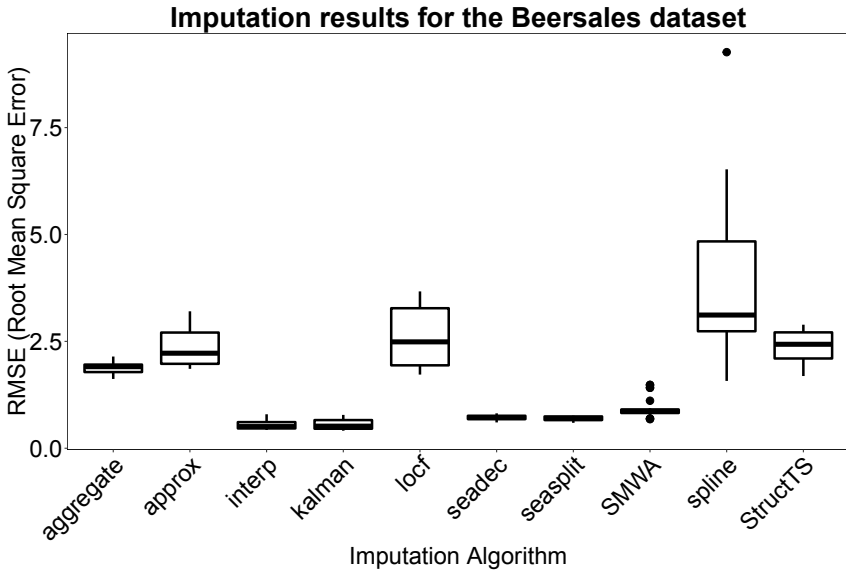


Figure 5: Imputation result for the Beersales dataset with 10% of missing data at random locations based on 30 different seeds

Table 3: Comparison of RMSE (mean) & Computation time (mean)- Beersales dataset for various imputation techniques for 10% of missing data at random locations based on 30 different seeds. Smaller values are better. Best values are shown in boldface.

Imputation Method	RMSE	Computation time (s)
Mean Imputation	1.88	0.001
Seasonal split	0.69	0.01
Seasonal decomposition	0.72	0.03
Spline interpolation	3.85	0.009
LOCF	2.61	0.008
STL based interpolation	0.55	0.05
Linear interpolation	2.33	0.012
SMWA imputation	0.93	0.42
Structural time series model	2.37	1.17
Kalman smoothing	0.55	4.61

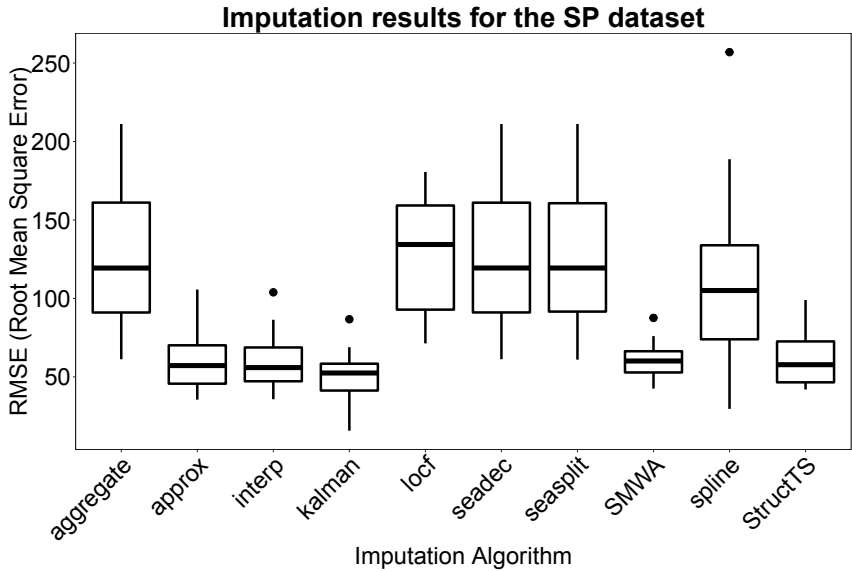


Figure 6: Imputation result for the SP dataset with 10% of missing data at random locations based on 30 different seeds

Table 4: Comparison of RMSE (mean) & Computation time (mean) - SP dataset for various imputation techniques for 10% of missing data at random locations based on 30 different seeds. Smaller values are better. Best values are shown in boldface.

Imputation Method	RMSE	Computation time (s)
Mean Imputation	124.16	0.001
Seasonal split	124.36	0.001
Seasonal decomposition	124.17	0.02
Spline interpolation	106.45	0.008
LOCF	129.55	0.009
STL based interpolation	59.76	0.03
Linear interpolation	59.89	0.01
SMWA imputation	60.40	0.34
Structural time series model	60.42	0.22
Kalman smoothing	49.78	1.01

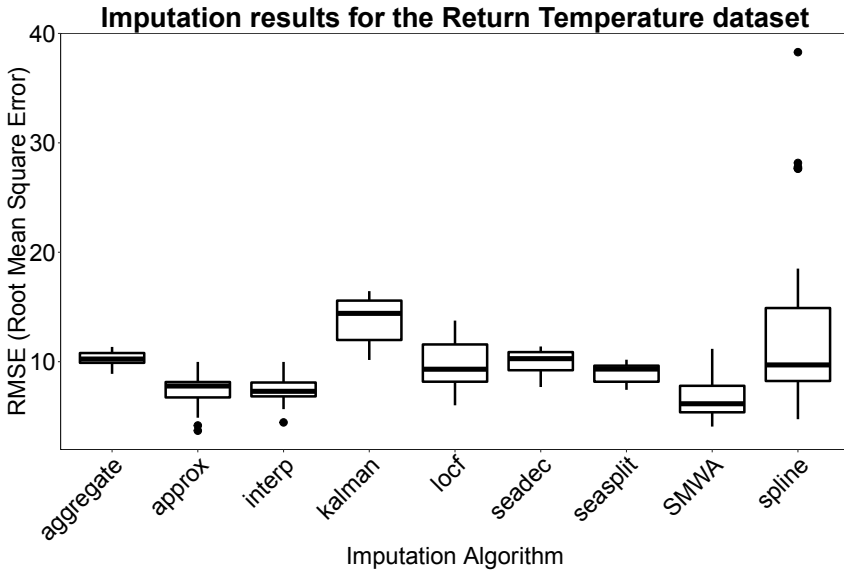


Figure 7: Imputation result for Return Temperature dataset with missing gap of size 100 at 10 random locations based on 30 different seeds

3.4 Case III: Large real-world data set with increased data gaps

In this test case, the complete *Return Temperature* dataset from the GECCO Industrial challenge is taken and 100 continuous data are removed randomly at 10 different locations for 30 runs. This particular scenario occurs in industries when data transmission or a sensor fails. The parameter settings for SMWA are $l = 100$, $g = 100$, and $w = 20,000$. Then SMWA is used to impute these large missing gaps along with other algorithms. The metrics in Table 5 clearly show the importance and performance of SMWA for very large industrial datasets. The SMWA bags the first position with a very low mean RMSE of 6.8 as in Figure 7. The Kalman Smoothing which served best in smaller data sets cannot perform well with larger intervals of missing data although it took very high computational time. Also, STL based interpolation, which showed better performance in smaller datasets ranks in second position following SMWA. It is to be noted that structural time series model is not evaluated with such large datasets for multiple runs as it takes very large processing time. In general, the performance of linear interpolation, STL based interpolation and SMWA are highly convincing

Table 5: Comparison of RMSE (mean) & Computation time (mean) - Return Temperature dataset for various imputation techniques with missing gap of size 100 at 10 random locations based on 30 different seeds. Smaller values are better. Best values are shown in boldface.

Imputation Method	RMSE	Computation time (s)
Mean Imputation	10.25	0.02
Seasonal split	9.02	1.45
Seasonal decomposition	9.97	3.25
Spline interpolation	13.47	1.49
LOCF	9.73	0.45
STL based interpolation	7.53	3.63
Linear interpolation	7.35	0.46
SMWA imputation	6.8	68.28
Kalman smoothing	13.87	4110.01
Structural time series model	NA	NA

in terms of RMSE with such large intervals of missing data. Though linear interpolation gives similar RMSE values compared to SMWA, they replace long gaps with a straight line, while SMWA tries to reproduce the similar complex pattern as in underlying real data. This pattern recovery combined with better accuracy and relatively less computational time makes the SMWA technique suitable for real time industrial data.

4 Conclusion

The SMWA is specially proposed for large intervals of missing data, which is a frequently occurring scenario in real industrial applications. The proposed SMWA combines good imputation accuracy with quick computational time. It focuses on extracting the best possible patterns from the available past data and utilizing it filling in the missing interval. The additional positive effect of using SMWA technique is for the ability of the algorithm to preserve the underlying real patterns better than other techniques.

It is also shown that the SMWA algorithm is well suited to work with various kinds of univariate datasets. Although algorithms like Kalman smoothing are highly robust for smaller datasets, they fail to perform

well with large intervals of missing data. Also, due to very expensive computational time, Kalman smoothing cannot be used in practice for large industrial datasets.

Furthermore, the SMWA technique can be utilized to fit the missing data even from the available future time series. This can be easily done by running the algorithm with a reversed time series. Further improvement could be gained by approximating the results of both, past and future data. This algorithm can be easily used on top of any other imputation algorithm, especially for large missing intervals.

Acknowledgement

Parts of this work have been developed in the project '*IMProvT: Intelligente Messverfahren zur Prozessoptimierung von Trinkwasserbereitstellung und -verteilung*' (reference number: 03ET1387A). Kindly supported by the Federal Ministry of Economic Affairs and Energy of the Federal Republic of Germany.

References

- [1] Imhoff, M., Bauer, M., Gather, U. and Löhlein, D. "Statistical pattern detection in univariate time series of intensive care on-line monitoring data." *Intensive care medicine* 24(12), pp.1305-1314. 1998.
- [2] Momani, P.E.N.M. and Naill, M. "Time series analysis model for rainfall data in Jordan: Case study for using time series analysis." *American Journal of Environmental Sciences*, 5(5), p.599. 2009.
- [3] Woolrich, M.W., Ripley, B.D., Brady, M. and Smith, S.M. "Temporal autocorrelation in univariate linear modeling of FMRI data." *Neuroimage*, 14(6), pp.1370-1386. 2001.
- [4] Taylor, J.W. "A comparison of univariate time series methods for forecasting intraday arrivals at a call center." *Management Science*, 54(2), pp.253-265. 2008.

- [5] Cuaresma, J.C., Hlouskova, J., Kossmeier, S. and Obersteiner, M. "Forecasting electricity spot-prices using linear univariate time series models." *American Journal of Environmental Sciences*, 5(5), p.599. 2004.
- [6] Stock, J.H. and Watson, M.W. "A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series." (No. w6607). National Bureau of Economic Research. 1998.
- [7] Moritz, S., Sardá, A., Bartz-Beielstein, T., Zaefferer, M. and Stork, J. "Comparison of different Methods for univariate Time Series Imputation in R." arXiv preprint arXiv:1510.03924. 2015.
- [8] Hyndman, R.J. and Khandakar, Y. "Automatic time series for forecasting: the forecast package for R (No. 6/07)." Monash University, Department of Econometrics and Business Statistics. 2007.
- [9] Rubin, D.B. "Inference and missing data." *Biometrika* 63(3), 581-592. 1976.
- [10] Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I. "Stl: A seasonal trend decomposition procedure based on loess." *Journal of Official Statistics* 6(1), 3-73. 1990.
- [11] Zeileis, A. and Grothendieck, G. "zoo: S3 infrastructure for regular and irregular time series." *Journal of Statistical Software* 14(1). 2005.
- [12] Moritz, S. "imputeTS: Time Series Missing Value Imputation." R package version 0.4. 2015.
- [13] Chan, K.S., Ripley, B. "Tsa: Time series analysis." R package version 1. 2012.
- [14] Brown, M.L. and Kros, J.F. "The impact of missing data on data mining." *Data mining: Opportunities and challenges*, 1, pp.174-198. 2003.
- [15] Ratanamahatana, C.A. and Keogh, E. "Everything you know about dynamic time warping is wrong." *Third Workshop on Mining Temporal and Sequential Data*. 2004.

Zum optimalen Testsignalentwurf für die Parametrierung von Partitionierung und Teilmodellen dynamischer Takagi-Sugeno-Modelle: Problemstellung und Lösungsansätze

Andreas Kroll

Fachgebiet Mess- und Regelungstechnik,
Institute for System Analytics and Control (ISAC),
Fachbereich Maschinenbau, Universität Kassel
Mönchebergstrasse 7, 34125 Kassel
E-Mail: andreas.kroll@mrt.uni-kassel.de

1 Einführung

Für den Testsignalentwurf zur Identifikation linearer dynamischer Modelle wurden verschiedene Lösungen entwickelt, die insbesondere die kompakte Parametrierbarkeit der Testsignale im Frequenzbereich ausnutzen [1]. Eine direkte Übertragung für den Testsignalentwurf zur Identifikation nichtlinearer dynamischer Modelle ist nicht möglich, da die Amplitudenabhängigkeit des Systemverhaltens dabei nicht berücksichtigt wird. In der Tat gilt das Problem des Testsignalentwurfs für allgemeine nichtlineare dynamische Modelle als nicht gelöst. Dies schließt auch die Klasse der dynamischen Takagi-Sugeno-Modelle ein, die im Zentrum dieses Beitrags steht.

Üblicherweise werden für die Identifikation dynamischer Takagi-Sugeno-Modelle Standardtestsignale wie amplitudenmodulierte binäre Pseudo-Rauschsignale (APRBS) oder Multi-Sinussignale eingesetzt. Diese werden i.d.R. intuitiv parametriert; Ansätze in Richtung einer systematischen Auslegung finden sich in [2, 3]. Ein prozessmodellfreier Entwurf führt prinzipbedingt nicht zu Testsignalentwürfen, die im Sinne von Prozessmodell-Gütemaßen optimal sind. Es gibt wenige Arbeiten zum prozessmodellbasierten Testsignalentwurf. Diese nehmen zudem wesentliche Vereinfachungen vor. So werden in [4] im Rahmen eines Echtzeitentwurfverfahrens lokal D-optimale Testsignale für den Fall einer gegebenen Partitionierung berechnet. Dabei soll insbesondere die Einhaltung von Wertebereichsgrenzen

von Prozessgrößen berücksichtigen werden. Eine ausführlichere Diskussion des Stands der Wissenschaft und Technik findet sich in [5, 6].

In diesem Beitrag wird der Testsignalentwurf für die Identifikation dynamischer zeitdiskreter Takagi-Sugeno-Modelle sowohl mit optimaler Partitionierung als auch mit optimalen Teilmodellparametern adressiert. Das resultierende Optimierungsproblem ist inaktabel. Deshalb gilt es Approximationen zu ermitteln, die das Problem einerseits berechenbar machen aber andererseits nur zu geringen Einbußen bei der Optimalität der Lösung führen. Der Beitrag stellt verschiedene Vorgehensweisen vor. Insbesondere wird ein Lösungsansatz beschrieben, dessen Idee darin besteht, die spezifische mathematische Struktur von Takagi-Sugeno-Modellen für eine Dekomposition in traktable Teilprobleme auszunutzen.

Der folgende Abschnitt enthält eine mathematische Fassung der Problemstellung. Es schließt sich ein Abschnitt über den Lösungsansatz an. Anschließend wird der Beitrag zusammengefasst und ein Ausblick gegeben. Beim Beitrag handelt es sich um eine konzeptionelle Arbeit, deren Konkretisierung, Umsetzung und Demonstration noch ausstehen. Es soll ein interessantes Problemfeld vorgestellt werden, das verschiedene Computational Intelligence-(CI-)Methoden involviert.

2 Problemstellung

2.1 Zielsystem und Modellansatz

Betrachtet werde der Testsignalentwurf für die Identifikation zeitdiskreter dynamischer Takagi-Sugeno-Modelle für zeitinvariante MISO-Systeme

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad y(t) = g(\mathbf{x}(t), \mathbf{u}(t)) \quad (1)$$

mit $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^p$ und $y \in \mathbb{R}$. Die Beobachtungen der Ausgangsgrößen $y_p(t)$ seien gestört, was als durch jeweils eine additive Zufallsvariable ϵ beschreibbar angenommen sei:

$$y_p(t) = y(t) + \epsilon(t) \quad (2)$$

Dabei sollen Takagi-Sugeno-Modelle mit linear-affinen Teilmodellen

$$y_i(k, \Theta_{LM,i}) = -a_{i,1} \cdot y(k-1) - \dots - a_{i,n} \cdot y(k-n)$$

$$\begin{aligned}
& + b_{i,1,1} \cdot u_1(k-1) - \dots - b_{i,m_1,1} \cdot u_1(k-m_1) + \dots \\
& + b_{i,1,p} \cdot u_p(k-1) - \dots - b_{i,m_p,p} \cdot u_p(k-m_p) + c_i \\
= & \Theta_{\text{LM},i} \cdot [\varphi^\top, 1]^\top \tag{3}
\end{aligned}$$

(ggf. zzgl. Totzeiten, die zwecks knapperer Notation in diesem Beitrag weggelassen werden) und Zugehörigkeitsfunktionen vom Typ Fuzzy-c-Means

$$\mu_i(k, \Theta_{\text{MF}}) = \left[\sum_{j=1}^c \left(\frac{\|\mathbf{z} - \mathbf{v}_i\|^2}{\|\mathbf{z} - \mathbf{v}_j\|^2} \right)^{\frac{1}{\nu-1}} \right]^{-1} \quad \text{mit } \Theta_{\text{MF}}^\top = [\mathbf{v}_1^\top, \dots, \mathbf{v}_c^\top] \tag{4}$$

betrachtet werden. Dabei ist k die diskrete Zeit, c die Anzahl der Teilmodelle/Partitionen, $\nu \in \mathbb{R}^{>1}$ der Unschärfeparameter, $\mathbf{v}_i/\mathbf{v}_j$ sind die Cluster- bzw. Partitionszentren und φ ist die Regressions- sowie \mathbf{z} die Scheduling-Variable. $\Theta_{\text{LM},i}$ enthält alle Parameter ($a_{i,j}, b_{i,j,l}, c_i$) des i -ten Teilmodells. Durch gewichtete Überlagerung der Teilmodelle entsteht das Gesamtübertragungsverhalten des Takagi-Sugeno-Modells gemäß:

$$y(k, \Theta_{\text{MF}}, \Theta_{\text{LM}}) = \sum_{i=1}^c \phi_i(k, \Theta_{\text{MF}}) \cdot y_i(k, \Theta_{\text{LM},i}) \quad \text{mit} \tag{5}$$

$$\Theta_{\text{LM}}^\top = [\Theta_{\text{LM},1}^\top, \dots, \Theta_{\text{LM},c}^\top] \quad \text{und} \quad \phi_i(k, \Theta_{\text{MF}}) = \frac{\mu_i(k, \Theta_{\text{MF}})}{\sum_{j=1}^c \mu_j(k, \Theta_{\text{MF}})}.$$

2.2 Identifikation

Für die Partitionierung werde der Fuzzy-c-Means (FCM) eingesetzt, bspw. mit Euklid'scher Abstandsnorm. Er liefert direkt die Zugehörigkeitsfunktionen. Dazu wird ν geeignet gewählt, siehe bspw. [7, 8]. Die Anzahl c an Partitionen kann durch Clustervalidierung oder Bewertung des Approximationsfehlers des resultierenden Takagi-Sugeno-Modells ermittelt werden. Die Parameter der Teilmodelle können durch die globale oder lokale Methode der kleinsten Quadrate geschätzt werden, siehe [7, 6]. Dies führt zu einem NARX-Modell. Nachgelagert kann das Modell auf seinen Simulationsfehler hin optimiert werden, was ein nichtlineares Schätzproblem bedeutet.

Damit sind die beim optimalen Testsignalentwurf zu berücksichtigenden Parameter die c zum Vektor Θ_{MF} zusammengefassten Partitionszentren und die zu den c Vektoren $\Theta_{\text{LM},i}$ zusammengefassten Parameter der Teilmodelle.

2.3 Optimaler Testsignalentwurf

Das Ziel eines optimalen Testsignalentwurfs besteht in der Berechnung einer Stellgrößenfolge $U = \{u(k)\}$ mit N Elementen, so dass ein Bewertungsmaß B für ein Experiment E minimiert wird. Dabei tritt die Systemdynamik, beschrieben über das Modell f^* , als Nebenbedingung auf:

$$U|_{0:N-1} = \arg \min_{u(0), \dots, u(N-1)} B(E) \text{ s.t. } y(k) = f^*(y(k-1), \dots, y(k-n), \\ u_1(k-1), \dots, u_1(k-m_1), \dots, u_p(k-1), \dots, u_p(k-m_p)) . \quad (6)$$

Experimententwürfe, die die Unsicherheit der geschätzten Modellparameter adressieren, nutzen i.d.R. die Cramer-Rao-Ungleichung aus, die besagt, dass bei jedem erwartungstreuen Schätzer die Kovarianzmatrix der Schätzwerte mindestens so groß wie die inverse Fisher-Informationsmatrix (FIM) ist. Die FIM ist über die Likelihood-Funktion $L(\Theta)$ definiert:

$$\mathbf{I}(\Theta) = [\mathbf{I}(\Theta)_{i,j}] \quad \text{mit} \quad \mathbf{I}(\Theta)_{i,j} = -\mathbb{E} \left\{ \frac{\partial^2 \ln L(\Theta|\mathbf{y})}{\partial \theta_i \cdot \partial \theta_j} \right\}_{\Theta=\Theta_0} \quad (7)$$

Dabei ist $\Theta^\top = [\Theta_{\text{MF}}^\top, \Theta_{\text{LM}}^\top]$ der Modellparametervektor und Θ_0 sein Referenzwert. $\mathbb{E}\{\cdot\}$ liefert den Erwartungswert einer Zufallsvariable. Im Fall unabhängiger, normalverteilter Zufallsvariablen ϵ mit gleicher Varianz σ^2 (i.i.d.), also wenn für die Kovarianzmatrix der Zufallsvariablen $\Sigma = \sigma^2 \cdot \mathbf{I}_N$ gilt (wobei \mathbf{I}_N eine $N \times N$ -Einheitsmatrix bezeichnet), vereinfacht sich die FIM im Fall von N Beobachtungen zu:

$$\mathbf{I} = \frac{1}{\sigma^2} \sum_{k=1}^N \frac{\partial y(k, \Theta)}{\partial \Theta} \cdot \frac{\partial y(k, \Theta)^\top}{\partial \Theta} \Big|_{\Theta=\Theta_0} \quad (8)$$

Bei LiP-Modellen hängt \mathbf{I} nicht von den Modellparametern ab und ein global optimaler Entwurf ist erzielbar. Bei Nicht-LiP-Modellen hängt das Entwurfsergebnis von den Modellparameterwerten ab und man spricht von einem „lokalen“ Entwurf. Praktisch werden die Zufallsgrößen oft nicht normalverteilt und oft auch nicht voneinander unabhängig sein, so dass der vereinfachte Ausdruck (8) nicht mehr gültig ist. Für allgemein normalverteilte Beobachtungen kann ein symbolischer Ausdruck für die FIM hergeleitet werden wobei dieser komplex ist, siehe bspw. [10]. Zudem müssen auch Mittelwert und Kovarianzmatrix der Verteilung bekannt sein, um die FIM berechnen zu können.

Die einfache Formel (8) ist aber auch ohne statistische Interpretation ein geeignetes Bewertungsmaß für die Nützlichkeit der Entwurfsunkte: Die Matrix ist aus den Parametersensitivitätsfunktionen aufgebaut; ausgewertet an allen Entwurfsunkten. Dabei zeigt die Auswertung für jeden Entwurfsunkt an, wie stark für ihn die Modellausgabe von den einzelnen Parametern beeinflusst wird. Beispielsweise liefert ein Entwurfsunkt im nahezu konstanten Sättigungsbereich einer sigmoiden Funktion nur wenig Informationen über die Funktionsparameter. Im Gegensatz dazu sind Punkte in Bereichen großer Änderung des Verhaltens (also großer Werte der Ableitungen) sehr informativ; hier beeinflussen die Funktionsparameter stark die Abhängigkeit zwischen Argument und Bild der Funktion. Eine Maximierung der Matrix \mathbf{I} bedeutet dann also, dass die Entwurfsunkte so gewählt werden, dass sich die Wirkung der Funktionsparameter bestmöglich in den Beobachtungen zeigt.

Zur Erreichung eines skalaren Optimierungsziels wird ein Bewertungsmaß M auf \mathbf{I} angewendet, das diese auf einen einfacher optimierbaren Skalar abbildet. Im Bereich der FIM-basierten Entwürfe bewerten typische Maße die Determinante, die Spur oder den kleinsten Eigenwert von \mathbf{I} . Die Optimierung dieser Maße führt auf das kleinste (Hyper-)Volumen, die kleinste Diagonallänge des einhüllenden (Hyper-)Quaders bzw. die geringste Exzentrizität des Konfidenz(-hyper-)ellipsoids der Parameterschätzwerte.

2.4 Einige Schlussfolgerungen für Takagi-Sugeno-Modelle

Parameteranzahl

Die Anzahl der beim Testsignalentwurf für Takagi-Sugeno-Modelle zu berücksichtigenden Parameter ist groß im Vergleich zum Testsignalentwurf zur Parametrierung („Kalibrierung“) theoretischer Modelle: In der Literatur ist bei letzteren unter einer „hohen Anzahl“ typischerweise eine einstellige Zahl zu verstehen [11, 12, 13]. Bei der Identifikation zeitdiskreter empirischer Modelle treten dagegen viele historische Modellterme auf. Deren Anzahl vervielfältigt sich bei Takagi-Sugeno-Modellen in Folge der Aufteilung in Teilmodelle, wobei zudem noch die Parameter für die Partitionierung hinzukommen. Die Parameteranzahl addiert sich insgesamt auf zu $c \cdot (\dim(\mathbf{v}_i) + \dim(\boldsymbol{\varphi}_i) + 1) = \dim(\Theta)$ (wobei die „1“ aus dem affinen Term resultiert). Im Beispiel des einfachen elektromechanischen Stellers aus [14] mit einer Ein- und einer Ausgangsgröße wurden drei Terme und acht Regeln verwendet, wobei der Scheduling-Raum um eine Dimension gegenüber dem

Regressor-Raum reduziert wurde, woraus insgesamt 48 Modellparameter resultierten. Im Beispiel des Gastraktes eines Dieselmotors als Mehrgrößensystem in [15] sind es sogar 105 Modellparameter. Damit wird eine FIM für Takagi-Sugeno-Modelle vergleichsweise groß. Zudem ist sie wegen der verschiedenen involvierten Parametergruppen oft schlecht konditioniert [5]. Dies erschwert ein FIM-basiertes optimales Testsignaldesign.

LiP- und nicht-LiP-Teilentwürfe

Der Aufbau von Takagi-Sugeno-Modellen aus mehreren Teilmodellen, die lokal gültig sind, hat verschiedene Konsequenzen für den Experimententwurf: So muss die Anzahl der Teilmodelle festliegen (aus Vorwissen oder mittels eines vorausgehenden Experiments abgeleitet), um die FIM berechnen/optimieren zu können. Ist bei einem Takagi-Sugeno-Modell vom Typ NARX die Partitionierung bekannt, so hängt die FIM nicht von Parameterwerten ab, d.h. das Testsignalentwurfproblem ist Linear-in-den-Parametern (LiP). Ist die Partitionierung dagegen unbekannt, so ist das Entwurfproblem zwar LiP bzgl. der lokalen Modellparameter, aber nicht-LiP bzgl. der Zugehörigkeitsfunktionsparameter [5]. Ist ein Takagi-Sugeno-Modell vom Typ NOE, so ist das Entwurfproblem auch bei bekannter Partitionierung nicht-LiP. Die Zugehörigkeitsfunktionen vom Typ FCM (4) sind stark nichtlinear und so sind es auch ihre partiellen Ableitungen nach den Parametern, die zur Konstruktion der FIM notwendig sind. Das bedeutet, dass FIM-basierte Bewertungsmaße i.d.R. multimodal sind.

Günstige Entwurfspunktlagen

Untersuchungen am Beispiel von Regressionsaufgaben zeigten [5], dass bei einem D-optimalen Entwurf für die Identifikation von Takagi-Sugeno-Modellen informative Lagen bzgl. der Teilmodellparameter auf den Rändern, insbesondere in den Ecken, der Partitionen liegen. Günstige Lagen bzgl. der Zugehörigkeitsfunktionsparameter liegen jeweils im Bereich der Schnittpunkte von innerer Partitions-grenze und der Verbindungslinie der benachbarten Partitionszentren. Dies illustriert ein einfaches Beispiel eines Takagi-Sugeno-Modells mit drei Teilmodellen:

$$\begin{aligned} y_1 &= -4 \cdot x_1 + 4 \cdot x_2 - 2 := [x_1; x_2; 1] \cdot \Theta_{LM,1} \\ y_2 &= 4 \cdot x_1 - 2 \cdot x_2 - 4 := [x_1; x_2; 1] \cdot \Theta_{LM,2} \\ y_3 &= 2 \cdot x_1 + x_2 + 1 := [x_1; x_2; 1] \cdot \Theta_{LM,3} \end{aligned}$$

und Zugehörigkeitsfunktionen vom Typ FCM (4) mit den Zentren in:

$$\mathbf{v}_1 = [0,5; 0,5]^\top, \mathbf{v}_2 = [0,5; 1,5]^\top, \mathbf{v}_3 = [1,5; 1,0]^\top.$$

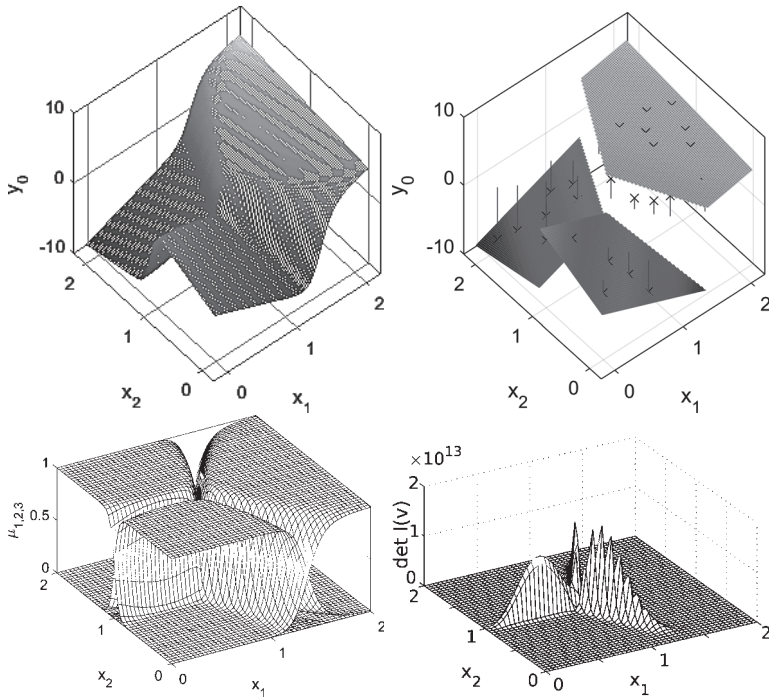


Bild 1: Takagi-Sugeno-Testsystem mit Graph (o.l.), Zugehörigkeitsfunktionen (u.l.), Teilmodellen (o.r.) und $\det \mathbf{I}$ in Abhängigkeit von der Platzierung eines der 9 Entwurfspunkte (u.r.).

Die Zugehörigkeitsfunktionen verwenden Euklid'schen Abstand und der Unschärfeparameter sei $\nu = 1,3$. Der zulässige Bereich der beiden Eingangsgrößen sei auf $x_j \in [0; 2], j = \{1; 2\}$ beschränkt. Bild 1 zeigt das Originalsystem (o.l.) mit Zugehörigkeitsfunktionen (u.l.) und Teilmodellen (o.r.). Für eine erste Untersuchung werde ein Referenzdatensatz mit 9 Punkten erzeugt, die näherungsweise in den Partitionsecken liegen: $\mathbf{X} = \{(0,05; 0,05); (1,275; 0,05); (0,05; 0,95); (0,05; 1,05); (1,275; 1,95); (0,05; 1,95); (1,95; 1,95); (1,95; 0,05); (0,925; 1)\}$. Zur isolierten Betrachtung werde angenommen, dass die Teilmodellparameter bekannt seien. Betrachtet werde nun die Änderung von $\det \mathbf{I}$, wenn der erste Entwurfspunkt innerhalb des zulässigen Wertebereichs $\mathbf{x} \in [0; 2] \times [0; 2]$ variiert wird. In Bild 1 (u.r.) ist die zuvor beschriebene optimale Lage zu erkennen. Eine ausführlichere Analyse findet sich in [5].

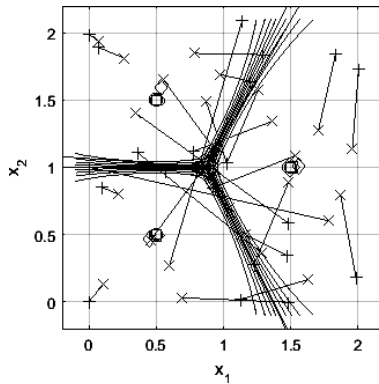


Bild 2: Entwurfspunkte (x) zu Beginn und nach Konvergenz (+) eines D-optimalen Entwurfs für das Takagi-Sugeno-Testsystem aus Bild 1. Die Partitionszentren (o/◇) ändern sich in diesem Beispiel nur minimal.

In einer zweiten Untersuchung wird für ein initiales Takagi-Sugeno-Modell für das o.a. Testsystem, das aus 25 mittels Latin-Hypercube-Sampling (LHS) platzierten Daten erzeugt wurde, ein D-optimaler Experimententwurf gleichzeitig bzgl. Teilmodell- und Zugehörigkeitsfunktionsparametern durchgeführt. Dieser verwendet die LHS-Entwurfspunkte als Startwerte. Bild 2 zeigt die initialen und die finalen Positionen der Entwurfspunkte. Es ist zu erkennen, dass die Punkte erwartungsgemäß zu den Partitions Grenzen wandern. Eine ausführlichere Analyse findet sich in [6].

Fazit

Optimale Entwürfe auf FIM-Basis erfordern Vorwissen aus einer vorausgehenden Experimental- und Identifikationsphase. In dieser werden die Anzahl der Teilmodelle und geeignete Initialparameter des Takagi-Sugeno-Modells ermittelt, um die FIM aufstellen und auswerten zu können. Hierzu wird typischerweise ein raumfüllender Experimententwurf eingesetzt, der den Testraum breit abdeckt. Das im Initialmodell enthaltene Vorwissen gestattet es dann, die relevanten Teile des Testraums eindeutig einzugrenzen und mit minimalem Experimentalaufwand einen maximalen Informationszugewinn für die Modellbildung zu erhalten. Es wurde gezeigt, dass die informativsten Orte bzgl. der Zugehörigkeitsfunktionsparameter auf den Partitions Grenzen liegen. Bei Systemen mit lokal sehr großen Gradienten oder sogar Unstetigkeiten müssen die Partitions Grenzen diese gut „treffen“, um große Modellfehler zu vermeiden. Bereits das Initialmodell sollte in

dieser Hinsicht ausreichend gut sein, so dass der FIM-basierte Entwurf zu Beobachtungen in tatsächlich relevanten Bereichen führt. Bei glatten nichtlinearen Systemen fallen Abweichungen dagegen nicht so stark ins Gewicht. Bei den Teilmodellparametern ist ein genauere Blick erforderlich: Bei einem glatten nichtlinearen System ist bzgl. der Teilmodellparameter zu unterscheiden, ob das Takagi-Sugeno-Modell auf sein globales Prädiktionsverhalten oder auf lokale Interpretierbarkeit hin ausgelegt werden soll. In ersterem Fall liegen die informativsten Orte wiederum auf den Partitions Grenzen. Im letzteren Fall ist zu beachten, dass sich im Übergangsbereich die Teilmodelle überlappen, was den Rückschluss auf einzelne Teilmodelle erschwert. Dazu kann es sinnvoll sein, Daten mit etwas Abstand zu den Partitions Grenzen zu erheben, wo Beobachtungen im Wesentlichen einem Teilmodell zugeordnet werden können. Dies kann bspw. durch die Einführung von Attraktivitätsfunktionen um die inneren Partitions Grenzen erreicht werden, die einen gewünschten Abstand favorisieren. Zur Schätzung der Modellparameter sollten i.d.R. alle verfügbaren Informationen genutzt werden, also auch die Daten aus der vorausgehenden Experimentalphase. Daten, die für die Teilmodellschätzung „ungünstig“ liegen (weil sie bspw. optimal bzgl. der Zugehörigkeitsfunktionsparameter platziert wurden), können bei der Schätzung geringer gewichtet werden.

Die gewählte Entwurfsvorgehensweise sollte sich also am Charakter des Zielsystems wie auch am Ziel der Modellbildung orientieren.

3 Lösungsansätze

Selbst bei einfachen Aufgaben wie der Identifikation nichtlinearer SISO-Modelle für mechatronische Stellglieder wie Drosselklappen [2, 14] oder für pneumatische Antriebe [16] finden i.d.R. multivariate Zeitreihen mit über 1000 Beobachtungen Einsatz. Bei komplexeren Aufgaben sind i.d.R. jenseits von 10.000 Beobachtungen typisch, siehe z.B. [15]. Jeden Stellgrößenwert als Freiheitsgrad zu behandeln führt zu intractablen Optimierungsproblemen. Diese Erkenntnis wurde früh bei linearen Modellen gesammelt. Deshalb erfolgt dafür i.d.R. eine parametrische Problemkodierung und ein Experimententwurf im Frequenzbereich. Dies ist für eine Identifikation nichtlinearer Modelle allerdings nicht angemessen, da die Amplitudenabhängigkeit des Systemverhaltens dabei unberücksichtigt bleibt. Deshalb stellt sich die Frage nach geeigneten Problemkodierungen.

3.1 Übersicht über Problemkodierungen

Bild 3 stellt den Versuch dar, eine systematische Übersicht über mögliche Vorgehensweisen bei der Kodierung von optimalen Testsignalentwurfproblemen zu geben⁴. Das bedeutet, dass prozessmodellfreie Methoden hier nicht berücksichtigt werden. Dabei sei angemerkt, dass unter einem „optimalen Experimententwurf“ in der Literatur i.d.R. ein Fisher-Informationmatrix-basierter Entwurf verstanden wird, der die Unsicherheit der Parameterschätzwerte bei der Identifikation minimieren soll. Dabei werden wie in Abschnitt 2.3 angesprochen i.d.R. vereinfachende Annahmen getroffen wie das Vorliegen unabhängiger, normalverteilter Zufallsvariablen gleicher Varianz, die die Berechnungen wesentlich vereinfachen. Kritisch ist anzumerken, dass in der Praxis die Wahrscheinlichkeitsverteilung deutlich von diesen Annahmen abweichen kann (bspw. wegen des Auftretens von statistischen Abhängigkeiten oder des Abweichens von der Normalverteilung). Es gibt auch Arbeiten, die Testsignale über die Minimierung einer approximationsfehler-orientierten Kostenfunktion entwerfen, ohne dabei spezielle statistische Annahmen zu treffen.

Die Offline-Entwürfe haben den Vorteil, dass die Berechnungen keinen harten Echtzeitanforderungen genügen müssen. Dadurch können aufwändigere Berechnungen durchgeführt und das gelöste Problem kann ähnlicher zum tatsächlichen Problem modelliert werden. Zu beachten ist, dass ein offline-entworfenen optimales Testsignal beim Auftreten von Modellfehlern (bspw. in Folge nichtmodellierter Dynamik, Parameterabweichungen oder weil das System nicht in der Modellklasse liegt) oder wegen nicht modellierten Störwirkungen zu suboptimalen Ergebnissen führen kann. Auch im Fall eines Offline-Entwurfs ist eine direkte Optimierung der einzelnen Elemente einer Stellgrößensequenz i.d.R. numerisch zu aufwändig, selbst wenn reduzierte Parametrierungen wie polygonzugartige Signale Verwendung finden. Beim Testsignalentwurf mittels Optimierung der Parameter eines ausgewählten Signaltyps/-modells wird der Lösungsraum entsprechend eingeschränkt. Bei den üblichen Standardtestsignalen wird der Multimodellaufbau der Takagi-Sugeno-Modelle weder direkt berücksichtigt noch ausgenutzt.

Letzteres ist dagegen bei den prozessmodellstruktur-orientierten Dekompositionsansätzen der Fall. Dieser Abschnitt fokussiert aus Platzgründen auf den Testsignalentwurf im Hinblick auf das Globalverhalten von Takagi-Sugeno-Modellen. Dabei wird die zuvor beschriebene Einsicht, dass bei

⁴Die Übersicht erhebt keinen Anspruch auf Vollständigkeit und stellt nur eine mögliche Strukturierung des Themenfeldes dar.

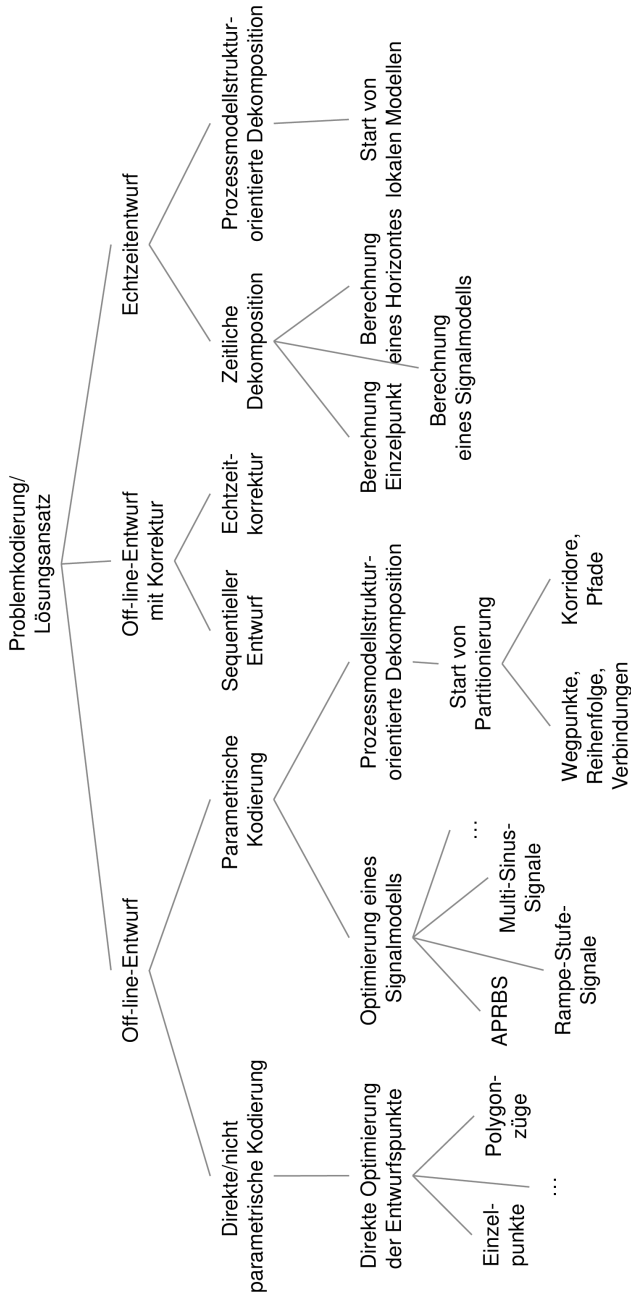


Bild 3: Strukturierte Übersicht über mögliche Ansätze zur Problemerkodierung optimaler Testsignal-Entwurfsverfahren für die Identifikation von Takagi-Sugeno-Modellen.

Takagi-Sugeno-Modellen die informativsten Orte an den Partitions-
grenzen liegen, zur approximativen Dekomposition des Entwurfsproblems ge-
nutzt: So kann aus der geschätzten Partitionierung ein Satz informativer
(Weg-)Punkte ermittelt werden, die dann mit einer Systemtrajektorie zu
verbinden sind. Alternativ können aus der Partitionierung Grenzkorridore
abgeleitet werden, die ein Testsignal ablaufen sollte.

Sind die geschätzten Grenzen der Partitionen unklar, aber gibt es einen
Satz von lokalen Modellen für verschiedene Betriebspunkte, so können
Testsignale auch dadurch entworfen werden, dass sie sich vom „Kernge-
biet“ der lokalen Modelle sukzessive auf die Partitions-
grenzen zu bewegen und bei deren Erreichen an diesen entlanglaufen. Dies muss im laufenden
Experiment erfolgen und gehört somit zu den Echtzeitentwürfen.

Die Echtzeittestsignalentwürfe unterliegen harten Echtzeitanforderungen,
weshalb i.d.R. wesentliche Vereinfachungen der Problemstellung durchge-
führt werden. So wird bspw. die Partitionierung als bekannt vorausgesetzt
und der Testsignalentwurf auf die Identifikation der lokalen Teilmodelle
beschränkt. Im Fall von (affinen) ARX-Teilmodellen sind die für die FIM
notwendigen partiellen Ableitungen dann von den Parameterwerten unab-
hängig. Beim Echtzeitentwurf wird typischerweise das Modell kontinuierlich
aktualisiert, so dass Modellfehler und Vernachlässigungen korrigiert werden.
Auch lässt sich durch Verringerung der Abweichung zwischen prädictiertem
und tatsächlichem Systemverhalten die Einhaltung von Beschränkungen
besser gewährleisten. Problemstellung und Lösungsansätze sind eng mit
der modellprädiktiven Regelung verwandt, wobei andere Kostenfunktionen
verwendet werden.

Die Offline-Entwürfe mit Korrektur sollen die Vorteile aus Offline- und Echt-
zeitentwurf verbinden und Nachteile kompensieren. Das bedeutet insbeson-
dere einen komplexeren Offline-Entwurf, der zu informativeren Testsignalen
führen kann, mit einem Echtzeitentwurf zu verbinden, der Modelldefizite
kompensieren kann. Korrekturen können auf verschiedene Art erfolgen: Bei
einem sequentiellen Entwurf (der zu den robusten Entwurfsverfahren gehört)
wird ein für eine Identifikation ausreichendes Testsignal (modellbasiert)
entworfen und auf den Prozess geschaltet. Mit den Beobachtungen wird
das Modell aktualisiert und auf dessen Basis ein neues Testsignal entworfen
und aufgeschaltet. Dies wird wiederholt, bis ein Abbruchkriterium erreicht
wird. Eine alternative Vorgehensweise besteht in der Korrektur mittels einer
Regelung. Dies kann auf verschiedene Weise umgesetzt werden, bspw. als
Tracking- oder Endpunktregler, was auch bedeutet, dass der korrigierende
Eingriff kontinuierlich oder auch nur temporär erfolgen kann.

Nur für einen Teil der beschriebenen Vorgehensweisen sind Arbeiten aus der Literatur bekannt. Wegen der Vielfalt der möglichen Vorgehensweisen wird im folgenden Abschnitt eine neuartige Vorgehensweise, die die spezifische Multi-Modellstruktur von Takagi-Sugeno-Modellen ausnutzt, in größerem Detail vorgestellt.

3.2 Behandlung als Pfadplanungsaufgabe

Zur Entwicklung von Strategien zur Problemdekomposition lässt sich die Teilmodellstruktur auszunutzen. Es wurde in Abschnitt 2.4 gezeigt, dass informative Orte für Takagi-Sugeno-Modelle mit linear-affinen Teilmodellen insbesondere auf den inneren und äußeren Partitions Grenzen liegen. Das motiviert ein Testsignalentwurfs- als ein Pfadplanungsproblem aufzufassen. Dies kann erfolgen, indem ein Pfad als Sequenz von Wegpunkten aufgefasst wird, die untereinander mit Teilpfaden verbunden werden. Daraus folgt erstens das kombinatorische Problem der Berechnung der optimalen Wegpunkt Reihenfolge, für das verschiedene CI-Methoden bereits erfolgreich eingesetzt wurden. Zweitens folgen Optimalsteuerungsprobleme, für die in der Regelungstechnik bspw. im Kontext der modellprädiktiven Regelung Lösungsmethoden entwickelt wurden.

Die Ermittlung der optimalen Reihenfolge von r Wegpunkten stellt ein Problem des Handlungsreisenden dar, das sich formulieren lässt als:

$$\{s_1, s_2, \dots, s_r\}_{\text{opt}} : \arg \min_{\{s_1, s_2, \dots, s_r\}} C_{s_r, s_1} + \sum_{j=1}^{r-1} C_{s_j, s_{j+1}} \quad (9)$$

Dabei sind die s_j die Wegpunkte, C bezeichnet eine Kostenfunktion und $C_{s_j, s_{j+1}}$ die Teilpfadkosten von Wegpunkt s_j zu Wegpunkt s_{j+1} . Eine Teilpfadentwurfsaufgabe kann als Optimalsteuerungsproblem gelöst werden. Bei einem nichtlinearen System können die Teilpfadkosten von der Laufrichtung abhängen. Zum Ermitteln einer vollständigen Teilpfadkostenmatrix sind bei r Wegpunkten $r^2 - r$ Optimalsteuerungsprobleme zu lösen; bei symmetrischen Pfadkosten die Hälfte. Für spärlich parametrisierte Modelle können die Berechnungskosten in einem akzeptablen Rahmen liegen. Bei größerer Wegpunktanzahl sind wegen des quadratischen Anstiegs der Berechnungskosten Vereinfachungen zu erwägen. Eine Möglichkeit besteht darin, die Teilpfadkosten für die Reihenfolgeplanung abzuschätzen und nur für die resultierende Reihenfolge Optimalsteuerungsprobleme zu lösen.

Dieser Ansatz kann allerdings zur Konvergenz in einen global suboptimalen Pfad führen. Ein einfacher und grober Ansatz für die Abschätzung besteht in der Verwendung von Wegkosten, die proportional zum Abstand der Wegpunkte im Scheduling- und/oder Regressorraum sind.

Wie lassen sich geeignete Wegpunkte festlegen?

Eine naheliegende Möglichkeit ist die Nutzung der Partitionszentren als Wegpunkte. Bei einer relativ direkten Verbindung zwischen den Zentren würde die Partitions Grenze allerdings im stumpfen Winkel traversiert. Dabei ist das Umfeld des Schnittpunktes zwischen Partitions Grenze und Verbindungslinie sehr informativ. Allerdings wird dann der größte Teil der Teilpfade innerhalb der Partitionen verlaufen, was nur wenig Information beisteuern würde. Bild 4 (o.r.) illustriert dies konzeptionell für ein Beispiel mit $c = 10$ Partitionen und entsprechend 10 Teilpfaden. Der dargestellte Verbindungspfad ist der mittels einer Kohonenkarte berechnete kürzest mögliche. Es sei angemerkt, dass so nicht gewährleistet wird, dass alle Partitions grenzen traversiert werden, was aber wünschenswert wäre. Alternativ können die Schnittpunkte zwischen Partitions Grenze und Verbindungslinien zwischen benachbarten Zentren als Wegpunkte verwendet werden. Da diese Schnittpunkte in den informativen Übergangsbereichen liegen werden Teilpfade, die in ihnen starten und enden, i.d.R. zu größeren Teilen im Übergangsbereich verlaufen. Bild 4 (u.l.) zeigt für das betrachtete Beispiel die Konstruktion der nun 21 Wegpunkte unter Nutzung einer Voronoi-Zerlegung für die gegebenen Partitionszentren. Bild 4 (u.r.) zeigt den mittels einer Kohonenkarte berechneten kürzesten Verbindungspfad zwischen allen Wegpunkten. Hier verlaufen die Teilpfade zu großen Teilen entlang der Übergangsbereiche, obwohl dies nicht im Optimierungsproblem modelliert wurde. (Es sei angemerkt, dass im Beispiel keine Systemdynamik beachtet wurde.)

Wie löst man die Optimalsteuerungsprobleme?

Bei der Lösung der Teilpfadplanungsaufgaben als Optimalsteuerungsproblem gilt es ein Testsignal in Form einer Sequenz diskreter Stellgrößen $U = \{u(k)\}$ so zu entwerfen, dass die Kostenfunktion

$$J = J_1(\mathbf{x}_e, k_e) + \sum_{k=k_0}^{k_e} J_2(\mathbf{x}_k, u_k, k), \quad \mathbf{x}(k_0) = \mathbf{x}_a, \quad (10)$$

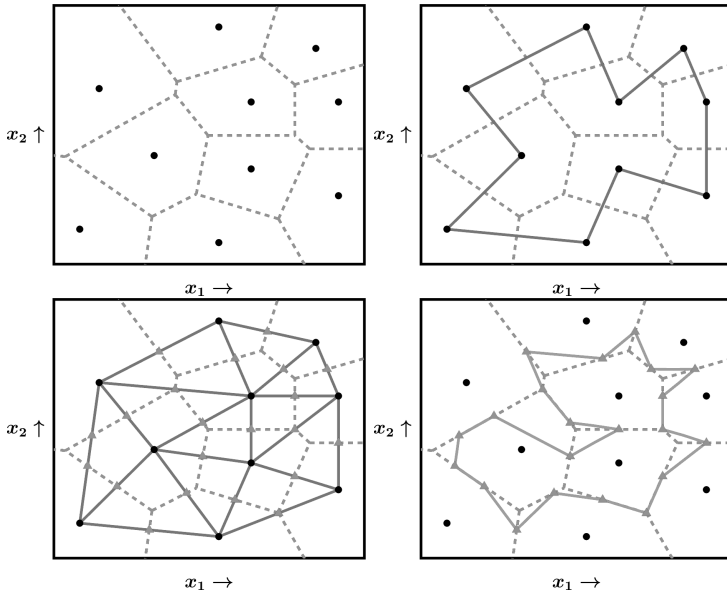


Bild 4: o.l.: Partitionierung mit $c=10$ Zentren (\bullet) und zugehörige Voronoi-Zerlegung (-); o.r.: Pfadlängenoptimierte Verbindung (—) zwischen den Partitionszentren (\bullet); u.l.: Ermittlung von Wegpunkten (\blacktriangle) aus den Schnittpunkten von Voronoi-Zerlegung (-) und Delaunay-Triangulation (-); u.r.: Pfad zwischen den Wegpunkten minimaler Gesamtlänge (—).

also die Summe aus Endpunkt- und Verlaufs- bzw. Pfadkosten unter Beachtung von Ungleichungs-Nebenbedingungen $h(\mathbf{x}_k, u_k, k) \leq 0$ für gegebenen Anfangswert \mathbf{x}_a minimal wird. Dabei sind J_1 , J_2 und h zeitinvariante Funktionale und $\mathbf{x}_e = \mathbf{x}(k_e)$ ist der End- und \mathbf{x}_a der Anfangspunkt der Zustandstrajektorie. (Als Zusatzziele können bspw. eine minimale Sequenzlänge ($k_e - k_0$), also ein möglichst kurzes Experiment und/oder ein ausgehnter Trajektorienverlauf im Übergangsbereich gefordert werden.) Eine einfache, im Bereich der Regelungstechnik übliche Konkretisierung stellt eine quadratische, zeitinvariante Kostenfunktion

$$\begin{aligned}
 J = & (\mathbf{x}_e - \mathbf{x}^*)^\top \cdot \mathbf{S} \cdot (\mathbf{x}_e - \mathbf{x}^*) \\
 & + \sum_{k=k_0}^{k_e} (\mathbf{x}_k - \mathbf{x}_{\text{soll},k})^\top \cdot \mathbf{Q} \cdot (\mathbf{x}_k - \mathbf{x}_{\text{soll},k}) + u_k^\top \cdot \mathbf{R} \cdot u_k \quad (11)
 \end{aligned}$$

mit dem Zielpunkt \mathbf{x}^* , der Solltrajektorie \mathbf{x}_{soll} und positiv (semi-)definiten Matrizen \mathbf{S} , \mathbf{Q} , und \mathbf{R} . (Bei linearen Systemen gilt $\mathbf{x}^* = \mathbf{x}_{\text{soll},k} = 0$.) Die Systemdynamik und Wertebereichsbeschränkungen wie $u_{\min} \leq u_k \leq u_{\max}$ oder $y_{\min} \leq y_k \leq y_{\max}$ treten als Nebenbedingungen auf. Dabei sind die Zielpunkte \mathbf{x}^* die zuvor ermittelten Wegpunkte. Das Ziel einer möglichst kurzen Experimentdauer lässt sich bspw. indirekt über eine mit k größer werdende Matrix \mathbf{R} modellieren.

Bei dieser Vorgehensweise wird die FIM nur indirekt optimiert und der Pfad wird i.d.R. nicht FIM-optimal sein. Für eine direkte Optimierung der FIM ersetzt man die o.a. Kostenfunktion einfach durch eine Metrik auf der FIM, wie bspw. der Determinante für einen D-optimalen Entwurf. Dem Vorteil einer besseren Modellierung des Entwurfsziels steht dann ein schwierigeres Optimierungsproblem gegenüber.

Für eine erste Untersuchung des Lösungsansatzes werde die Zielfunktion vereinfachend auf die Endpunktkosten $J = (\mathbf{x}_e - \mathbf{x}^*)^\top \cdot \mathbf{S} \cdot (\mathbf{x}_e - \mathbf{x}^*)$ reduziert. Das heißt es wird nur bewertet, wie nahe der erreichte Endpunkt \mathbf{x}_e am Zielpunkt \mathbf{x}^* liegt, nicht aber der Pfadverlauf selber. Bspw. gilt für ein lokal linear-affines Takagi-Sugeno-SISO-Modell vom Typ NARX:

$$y(k) = \sum_{i=1}^c \phi_i(k) \cdot (c_i + \mathbf{x}^\top(k-1) \cdot \Theta_{\text{LM},i}) \quad (12)$$

$$= \sum_{i=1}^c \phi_i(k) \cdot \left(c_i - \sum_{j=1}^n a_{i,j} \cdot y(k-j) + \sum_{j=1}^m b_{i,j} \cdot u(k-j) \right)$$

Vereinfachend sei angenommen, dass Scheduling- und Regressor-Variable gleich gewählt werden. Der $y(k_e)$ liefernde Entwurfspunkt ist $\mathbf{x}_e = \mathbf{x}(k_e - 1) = [-y(k_e - 1) \cdots -y(k_e - n)u(k_e - 1) \cdots u(k_e - m)]^\top$. Die Endpunktkosten sind bspw. für eine Wahl von $\mathbf{S} = \mathbf{I}$:

$$J = (\mathbf{x}_e - \mathbf{x}^*)^\top \cdot \mathbf{S} \cdot (\mathbf{x}_e - \mathbf{x}^*)$$

$$= \sum_{j=1}^n (-y(k_e - j) - x_j^*)^2 + \sum_{j=1}^m (u(k_e - j) - x_{n+j}^*)^2 \quad (13)$$

Im Fall einer direkten Optimierung besteht das Ziel dann darin, eine Stellgrößenfolge $U_{0:k_e-1} = \{u(0), \dots, u(k_e - 1)\}$ von N Werten zu ermitteln, die J minimiert. Wie schon in Abschnitt 3.1 erläutert können alternativ kompaktere Signalmodelle verwendet werden.

Zur Lösung des Optimierungsproblems können ableitungsbasierte Verfahren genutzt werden. Die Freiheitsgrade sind die N Stellgrößen $u(k)$. Die partiellen Ableitungen $\partial J/\partial u(k)$ können mittels der Kettenregel der Differenzialrechnung berechnet werden. Sie folgen bspw. zu:

$$\begin{aligned} \frac{\partial J}{\partial u(k)} &= 2 \cdot \sum_{j=1}^n (y(k_e - j) + x_j^*) \cdot \frac{\partial y(k_e - j)}{\partial u(k)} \\ &\quad + 2 \cdot \sum_{j=1}^m (u(k_e - j) - x_{n+j}^*) \cdot \frac{\partial u(k_e - j)}{\partial u(k)} \end{aligned} \quad (14)$$

Dabei entfallen alle Summanden der zweiten Zeile, für die $k_e - l \neq k$ gilt. Für die Ableitungsterme der ersten Zeile gilt:

$$\begin{aligned} \frac{\partial y(k_e - l)}{\partial u(k)} &= \sum_{j=1}^c \left[\frac{\partial \phi_i(k_e - l)}{\partial u(k)} \cdot \left(c_i - \sum_{j=1}^n a_{i,j} \cdot y(k_e - l - j) + \sum_{j=1}^m b_{i,j} \cdot u(k_e - l - j) \right) \right. \\ &\quad \left. + \phi_i(k_e - l) \cdot \left(\sum_{j=1}^n a_{i,j} \cdot \frac{\partial y(k_e - l - j)}{\partial u(k)} + \sum_{j=1}^m b_{i,j} \cdot \frac{\partial u(k_e - l - j)}{\partial u(k)} \right) \right] \end{aligned} \quad (15)$$

Bei der zweiten Summe in der zweiten Zeile entfallen alle Summanden, für die $k_e - l - j \neq k$ gilt. Die partiellen Ableitungen sind rekursiv zu berechnen, da die Ausgangsgröße von den zuvor aufgeschalteten Stellgrößen abhängt. Die Ableitung der Fuzzy-Basisfunktionen nach den Stellgrößen kann aufbauend auf die Ableitungen nach den Zentrumskomponenten erfolgen, führt aber zu sehr komplizierten Ausdrücken.

Lösung des Optimierungsproblems

Bei einem Prädiktionsfehler-orientierten optimalen Entwurf mit Hilfe von Gradientenverfahren sind erste partielle Ableitungen erforderlich. Insbesondere die partiellen Ableitungen der Zugehörigkeitsfunktionen sind kompliziert, wie bspw.:

$$\begin{aligned} \left. \frac{\partial y(k, \Theta)}{\partial \Theta_i} \right|_{\Theta_i = v_{i,j}} &= \sum_{i=1}^c \frac{\partial \mu_i(k, \Theta_{MF})}{\partial v_{i,j}} \cdot y_i(k, \Theta_{LM}) \quad \text{mit} \quad (16) \\ \frac{\partial \mu_i(k, \Theta_{MF})}{\partial v_{i,f}} &= \frac{-1}{\nu - 1} \cdot \left[1 + \sum_{\substack{j=1 \\ j \neq i}}^c \left(\frac{d_i^2}{d_j^2} \right)^{\frac{1}{\nu-1}} \right]^{-2} \cdot \sum_{\substack{j=1 \\ j \neq i}}^c \left(\frac{d_i^2}{d_j^2} \right)^{\frac{2-\nu}{\nu-1}} \cdot \frac{-2 \cdot (x_f - v_{i,f})}{d_j^2} \end{aligned}$$

und $d_i^2 = \|\mathbf{x} - \mathbf{v}_i\|_2^2$ und $d_j^2 = \|\mathbf{x} - \mathbf{v}_j\|_2^2$ im Fall Euklid'scher Abstände. Diese Ableitungen führen zu multi-modalen Optimierungsproblemen. Bei einem FIM-basierten Entwurf bedeutet der Einsatz von Gradientenverfahren, dass nochmals kompliziertere zweite partielle Ableitungen auftreten.

Zum einen sollten also Möglichkeiten zur Problemvereinfachung erwogen werden, wie eine geschickte Kodierung des Problems. Zum anderen muss die Multimodalität der Kostenfunktion, die auch bei Relaxation des Problems nicht vollständig verschwinden wird, adressiert werden. Letzteres kann mit globalen Suchverfahren der Computational Intelligence angegangen werden, wie Evolutionäre Algorithmen oder die Partikel-Schwarm-Optimierung. Diese Methoden gestatten es zudem, weitere Entwurfsanforderungen in der Zielfunktion mit zu modellieren.

4 Zusammenfassung und Ausblick

Im vorliegenden Beitrag wurde der optimale Testsignalentwurf für die Identifikation zeitdiskreter Takagi-Sugeno-Modelle für dynamische nichtlineare Systeme behandelt. Hierzu wurden die Besonderheiten des Modellansatzes und daraus resultierende Konsequenzen bzgl. des Entwurfsproblems untersucht. Eine strukturierte Übersicht über mögliche Problemkodierungen wurde vorgeschlagen, welche publizierte und neue Ansätze einordnet.

Ein bezüglich aller Entwurfspunkte optimales Testsignal ist für praktische Probleme i.d.R. nicht berechenbar. Der für lineare Modelle verbreitete Ansatz eines Entwurfs im Frequenzbereich ist für nichtlineare Systeme nicht ausreichend. So wurde zur Verringerung der Berechnungskomplexität vorgeschlagen, das Entwurfsproblem zu dekomponieren. Ein die spezielle Struktur von Takagi-Sugeno-Modellen ausnutzender Ansatz besteht in der Behandlung des Testsignalentwurfs- als Pfadplanungsproblem und der Nutzung der jeweils dafür entwickelten speziellen Lösungsansätze. Dazu erfolgt eine Zerlegung in das kombinatorische Problem der optimalen Reihenfolge von Wegpunkten und das (ggf. beschränkt) kontinuierliche Problem optimaler Teilpfadverläufe. Ersteres gehört zur Klasse der Probleme des Handlungsreisenden, für das verschiedene CI-Methoden zu guten Ergebnisse geführt haben. Letzteres gehört zur Klasse der Optimalsteuerungsprobleme, für die in der Regelungstechnik Lösungsmethoden erarbeitet wurden. Aus der speziellen Struktur eines Multi-Modells können dabei geeignete Wegpunkte wie auch andere Vereinfachungen abgeleitet werden. Dabei wird vorausgesetzt,

dass aus einer vorausgehenden Experimental- und Identifikationsphase ein initiales TS-Modell verfügbar ist.

Die Aufgabenstellung bietet ein interessantes interdisziplinäres Anwendungsfeld für CI-Methoden: Beim Takagi-Sugeno-Modell handelt es sich um einen Fuzzy-Ansatz. Für die Lösung des NP-harten Reihenfolgeoptimierungsproblems bieten sich Methoden bspw. aus den Bereichen der Evolutionären Algorithmen und der Schwarmintelligenz an. Dies gilt auch für die Lösung des Optimalsteuerungsproblems. Der Schlüssel für die Lösung scheint in einer geeigneten Problemkodierung zu liegen, zu der in der Literatur bisher nur erste Überlegungen existieren. Zukünftig gilt es die vorgeschlagenen Lösungsansätze auszuarbeiten und deren Leistungsfähigkeit in Fallstudien zu untersuchen.

Danksagung

Dank gilt Herrn Dipl.-Ing. Axel Dürrbaum für den Satz in L^AT_EX.

Literatur

- [1] G. C. Goodwin and R. L. Payne, „Dynamic system identification: experiment design and data analysis,“ New York, Academic Press, 1977.
- [2] M. Gringard and A. Kroll, „Zur Homogenisierung von Breitbandtestsignalen für die nichtlineare Systemidentifikation am Beispiel eines nichtlinearen Stellantriebs,“ in *Proceedings 25. Workshop Computational Intelligence*, GMA-FA 5.14, Dortmund, 26.–27. November 2015, pp. 145–162.
- [3] M. Gringard and A. Kroll, „Zur systematischen Analyse des Einflusses der Parametrierung von Standardtestsignalen für die Identifikation nichtlinearer dynamischer TS-Modelle am Beispiel eines elektromechanischen Stellantriebs,“ in *Proceedings 26. Workshop Computational Intelligence*, Dortmund, 24.–25. November 2016,
- [4] C. Hametner, M. Stadlbauer, M. Deregnaucourt, S. Jakubek, and T. Winsel, „Optimal experiment design based on local model networks

- and multilayer perceptron networks,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 251–261, 2013.
- [5] A. Kroll and A. Dürrbaum, „On joint optimal experiment design for identifying partition and local model parameters of Takagi-Sugeno models,” in *Proceedings 17th IFAC Symposium on System Identification (SysID)*, Beijing, China, 19.–21. Oktober 2015, pp. 1427 – 1432.
- [6] A. Dürrbaum and A. Kroll, „On robust experiment design for identifying locally affine Takagi-Sugeno models,” in *Proceedings IEEE International Conference on Systems, Man, and Cybernetics (SMC 2016)*, Budapest, Hungary, 9.–12. Oktober 2016.
- [7] A. Kroll, „Zur Modellierung unstetiger sowie heterogener nichtlinearer Systeme mittels Takagi-Sugeno-Fuzzy-Systemen,” in *Proceedings 20. Workshop Computational Intelligence*, Dortmund, 1.–3. Dezember 2010, pp. 64–79.
- [8] A. Kroll, „On choosing the fuzziness parameter for identifying TS models with multidimensional membership functions,” *Journal of Artificial Intelligence and Soft Computing Research*, vol. 1, no. 4, pp. 283–300, 2011.
- [9] O. Nelles, „Nonlinear system identification,” Springer, Berlin, 2001.
- [10] S. M. Kay, „Fundamentals of statistical signal processing, volume I: estimation theory,” Prentice Hall, Upper Saddle River, 1993.
- [11] C. P. Kitsos, *Optimal Experimental Design for Non-Linear Models: Theory and Applications*. Springer, Heidelberg, 2013.
- [12] G. Franceschini and S. Macchietto, „Validation of a model for biodiesel production through model-based experiment design,” *Industrial & Engineering Chemistry Research*, vol. 46, no. 1, pp. 220–232, 2007.
- [13] G. Franceschini and S. Macchietto, „Model-based design of experiments for parameter precision: State of the art,” *Chemical Engineering Science*, vol. 63, no. 19, pp. 4846–4872, 2008.
- [14] S. Zaidi and A. Kroll, „Electro-Mechanical Throttle as a Benchmark Problem for Nonlinear System Identification with Friction,” in *Proceedings 24. Workshop Computational Intelligence*, Dortmund, 27.–28. November 2014, pp. 173–186.

- [15] M. Kahl, A. Kroll, R. Kästner, and M. Sofsky, „Zur automatisierten Auswahl signifikanter Regressoren für die Identifikation eines dynamischen Ladedruckmodells,“ in *Proceedings 24. Workshop Computational Intelligence*, Dortmund, 27.–28. November 2014, pp. 33–53.
- [16] S. Zaidi and A. Kroll, „On Interval-valued-data Type-1 Takagi-Sugeno fuzzy systems for uncertain nonlinear dynamic system identification,“ *at – Automatisierungstechnik*, vol. 64, no. 6, pp. 418–427, 2016.

On the systematic parametrization of APRBS and multisine test signals for nonlinear system identification

Matthias Gringard, Andreas Kroll

Department of Measurement and Control,
Institute for System Analytics and Control (ISAC),
Faculty of Mechanical Engineering, University of Kassel
Mönchebergstraße 7, 34125 Kassel
E-Mail: {matthias.gringard, andreas.kroll}@mrt.uni-kassel.de

1 Introduction

Since the most powerful methods for simulation, prognosis and control design are model-based, high quality models are needed. Due to the ever rising complexity of technical systems as well as the need for nonlinear description of their components, theoretical modeling has become a difficult task that requires a lot of effort and expert knowledge. Therefore system identification is a more than valid alternative. However, identified models can in principle only describe the behavior that is present in the data used for identification and not be used for extrapolation like high quality physical models can. This directly leads to the task of generating suitable data for nonlinear system identification by exciting the system appropriately. Since Takagi-Sugeno-(TS-)Fuzzy models are universal approximators, they have been selected as the model class. The nonlinear behavior of locally affine TS-models is solely described by the partitioning, which emphasizes the importance of the structure identification.

The Design of Experiments (DoE) can be categorized in different ways: DoE can be done offline or in real time. If the design is done offline the complete system input signal will be designed before the experiment is conducted. In case of a real time design the input will be adapted during the experiment. Since there are computational difficulties with nonlinear model approaches, DoE can be categorized in this regard. In contrast to static systems where the design point can be chosen independently from each other, the system input of dynamic systems is a time series and the system dynamics have to be considered as a side condition. Test signal design is a part of DoE. The last major distinction is whether knowledge about the model and its

structure is used in the process. If the design is done without considering the model structure, it is called model-free, otherwise it is model-based. The model-based designs are typically based on the approach to optimize scalar measures on the Fisher Information Matrix (FIM). The FIM is constructed from the global model's first derivatives with respect to the model parameters. Exploiting the Cramér-Rao-Inequality the FIM is used to estimate the covariance of the parameters. Therefore the criteria can be interpreted as a way to minimize the model parameter uncertainty. The model partitioning and the dynamic order of the subsystems are often assumed to be known [1].

The membership functions that determine the weighting of the locally-affine models mostly depend nonlinearly on the partition prototypes. Since the system's nonlinearities are solely described by the membership functions, the prototypes (the parameters of the membership functions) have to be considered when designing excitation signals. While the FIM for linear models is independent of the parameters for linear models, in case of nonlinear model approaches the FIM depends on the local model parameters. They are however unknown at this stage of the experiment design. Experiments show that FIM-based design approaches are very sensitive to parameter mismatch [2], so a sequential robust design was proposed. Therefore the current research investigates the application of manipulated general purpose test signals for the identification of nonlinear TS-models. This model-free approach was used to generate broadband test signals that were manipulated to achieve a specific amplitude distribution either in the time or the frequency domain [3].

2 Model Family and Identification Process

As mentioned before, locally linear affine TS-models are used. TS-models are a superposition of local models weighted by their respective normalized membership functions. In case of (N)ARX-type models the i -th local model can be described by:

$$\hat{y}_i(k) = - \sum_{j=1}^{n_{yc}} a_{i,j} \cdot y(k-j) + \sum_{l=1}^{n_{uc}} b_{i,l} \cdot u(k-l) + c_i \quad (1)$$

$\hat{y}_i(k)$ is the output of the i -th local model, $a_{i,j}$ and $b_{i,l}$ are the coefficients of the in- and outputs as well as their lags, n_{yc} and n_{uc} are the maximum

lags, k is the discrete time and c_i is the affine term. The difference equation (1) can be rewritten as:

$$\begin{aligned} \hat{y}_i(k) &= \underbrace{\begin{bmatrix} 1 & \boldsymbol{\varphi}_y^T(k-1) & \boldsymbol{\varphi}_u^T(k-1) \end{bmatrix}}_{\boldsymbol{\varphi}^T(k-1)} \underbrace{\begin{bmatrix} c_i \\ \boldsymbol{\Theta}_{i,y} \\ \boldsymbol{\Theta}_{i,u} \end{bmatrix}}_{\boldsymbol{\Theta}_i} \\ &= \boldsymbol{\varphi}^T(k-1) \cdot \boldsymbol{\Theta}_i \end{aligned} \quad (2)$$

where $\boldsymbol{\varphi}(k-1)$ and $\boldsymbol{\Theta}_i$ are the regression vector and the i -th local model parameter vector. To obtain the global model, the local models (2) are weighted by their fuzzy basis functions $\phi_i(\mathbf{z}(k-1))$ and superposed. To emphasize the dependance of the regression vector and the scheduling variable $\mathbf{z}(k-1)$ from past values of the in- and output, the argument $(k-1)$ is carried along. The fuzzy basis functions are:

$$\phi_i(\mathbf{z}(k-1)) = \frac{\mu_i(\mathbf{z}(k-1))}{\sum_{j=1}^c \mu_j(\mathbf{z}(k-1))} \quad (3)$$

with

$$\mu_i(\mathbf{z}(k-1)) = \left[\sum_{j=1}^c \left(\frac{\|\mathbf{z}(k-1) - \mathbf{v}_i\|_2}{\|\mathbf{z}(k-1) - \mathbf{v}_j\|_2} \right)^{\frac{2}{\nu-1}} \right]^{-1} \quad (4)$$

in case of Fuzzy-c-means-type membership functions. The membership functions μ_i are functions of the scheduling variable $\mathbf{z}(k-1)$. The scheduling variable is often a function of the regression vector but can be chosen to be entirely different if a priori knowledge is available. c is the number of local models, the \mathbf{v}_i are the partitions' prototypes and $\nu \in \mathbb{R}^{>1}$ is the fuzziness parameter. Since the membership functions that are of the same type as in the Fuzzy-c-means (FCM) clustering algorithm fulfill the orthogonality condition,

$$\sum_{i=1}^c \mu_i(\mathbf{z}(k-1)) = 1 \quad (5)$$

the fuzzy basis functions are equal to the membership functions:

$$\phi_i(\mathbf{z}(k-1)) = \mu_i(\mathbf{z}(k-1)) \quad (6)$$

Therefore the global model can be given as:

$$\hat{y}(k|k-1) = \sum_{i=1}^c \mu_i(\mathbf{z}(k-1)) \cdot \boldsymbol{\varphi}^T(k-1) \cdot \boldsymbol{\Theta}_i \quad (7)$$

With the abbreviations $\mu_i(\mathbf{z}(k-1)) = \mu_{i,k}$ and $\boldsymbol{\varphi}^T(k-1) = \boldsymbol{\varphi}_k^T$ follows from (7):

$$\hat{y}(k|k-1) = [\mu_{1,k}\boldsymbol{\varphi}_k^T \quad \cdots \quad \mu_{c,k}\boldsymbol{\varphi}_k^T] \begin{bmatrix} \boldsymbol{\Theta}_1 \\ \vdots \\ \boldsymbol{\Theta}_c \end{bmatrix} \quad (8)$$

Considering N observations (8) is evaluated $(N - n_{yc})$ times:

$$\hat{\mathbf{y}} = \boldsymbol{\Phi}_E \boldsymbol{\Theta} \quad (9)$$

$\boldsymbol{\Phi}_E$ is the extended regression matrix, $\boldsymbol{\Theta}$ the vector of all local model parameters and $\hat{\mathbf{y}}$ the regressand. With known \mathbf{v}_i the estimation problem is linear in the parameters and the solution of the least squares problem can be given as follows when using a quadratic cost function for identification:

$$\hat{\boldsymbol{\Theta}} = (\boldsymbol{\Phi}_E^T \boldsymbol{\Phi}_E)^{-1} \boldsymbol{\Phi}_E^T \hat{\mathbf{y}} \quad (10)$$

The solution (10) is optimal with respect to the one-step-ahead-prediction (OSAP). To calculate it, c and \mathbf{v}_i have to be known. Therefore structural decisions like the selection of the scheduling variable, the number of local models and the type of membership function have to be made before the parameter estimation. The FCM algorithm is used to determine the prototypes \mathbf{v}_i . The resulting prototypes are not optimal with respect to the prediction error. The estimated parameter vector $\hat{\boldsymbol{\Theta}}$ minimizes the squared sum of the deviations of the one-step-ahead-prediction. For simulation purposes it is more important to have a model with good prediction quality when the model is evaluated recursively. This obviously is a harder criterion. Because of that the parameters are optimized with a nonlinear optimization algorithm with respect to the prediction error while the model is evaluated recursively.

The local model parameters $\boldsymbol{\Theta}_i$ and prototypes \mathbf{v}_i are aggregated to a parameter vector for the complete local model:

$$\boldsymbol{\Theta}_{\text{agg},i}^T = [\boldsymbol{\Theta}_i^T \quad \mathbf{v}_i^T] \quad (11)$$

The local model parameters and the partition are now optimized simultaneously. With a quadratic cost function the following holds:

$$\hat{\Theta}_{\text{agg}} = \underset{\Theta_{\text{agg}}}{\operatorname{argmin}} \sum_{k=1}^N (y_k - \hat{y}_k(\Theta_{\text{agg}}))^2 \quad (12)$$

with

$$\Theta_{\text{agg}}^{\text{T}} = [\Theta_{\text{agg},1}^{\text{T}} \quad \dots \quad \Theta_{\text{agg},c}^{\text{T}}] \quad (13)$$

The optimization problem is solved with the `MATLAB`-function `lsqnonlin` which by default uses a Trust-Region-Reflective-Algorithm. The optimization is initialized with the local model parameters from (10) and the prototypes from the FCM clustering. The choice of ν is discussed in detail in [4].

3 Design of the excitation signals

This section is divided into two parts. In the first part standard test signals are discussed. The second part is about how some of these signals can be parametrized to obtain better identification data.

3.1 Standard test signals

Standard test signals are used to excite a system without any prior knowledge. These signals are generally obtained by using two basic signal types. On the one hand there are signals based on sinusoids like swept sines or multisines. On the other hand multi-level signals are used. Furthermore different kinds of signals based on random noise are used

The crest factor for a signal x_k is defined as:

$$c_{\text{R}} = \frac{\max_{1 \leq k \leq N} |x_k|}{x_{\text{RMS}}} \quad (14)$$

The main difference between the design for linear and nonlinear system identification is that the amplitudes are neglected in the linear case since the choice of the amplitude levels does not yield any additional information. In the nonlinear case the amplitudes represent a degree of freedom that can be used to obtain a test signal that is better suited for identification.

3.2 Parametrization of test signals

As it was mentioned in the last section, multilevel and harmonic signals can be parametrized very comfortably, because the parameters can be interpreted easily. The signals that will be discussed here are multisine and APRBS signals. The parameters of a multisine are the used frequencies, the corresponding phases as well as the amplitudes. While the used frequencies are determined by the frequency band of interest, the choice of the phases and amplitudes are not that straightforward. When parametrizing a signal without special knowledge of the underlying system, goals can include to transfer as much energy into the system as possible. A uniform distribution in scheduling space might yield better partitioning since there is no particular weighting on certain regions. Due to the obvious lack of knowledge about the real system, these properties can only be approached indirectly. Without prior knowledge the starting point for the design is to achieve a uniform amplitude distribution. Since a signal in the time domain always faces certain boundaries on maximum (and minimum) values, the phases are used to achieve a distribution such that the amplitudes of the used frequencies can be scaled up to guarantee the maximal possible energy inserted into the system. A method for homogenization will be presented. For multilevel signals the impact of the parametrization is not that easy to interpret anymore. The values of the held input levels can be adjusted as well as the holding time. To analyse the impact of the holding time, experiments with the identical amplitude levels but varying holding times have been conducted. The amplitudes are chosen randomly.

3.2.1 Multisine Signals

The used homogenization method [5] iteratively adjusts the phases in the frequency domain until the signal in the time domain meets a criterion that is linked to the distribution. In this case a uniform distribution is used but the extension to arbitrary distributions is straightforward. The method has been presented in [3].

Figure 1 shows the impact of the homogenization of a full multisine with 50 frequencies between 0 Hz and 5 Hz and Schroeder phases [6]. It can be seen that both signals remain within the values of ± 1 . Furthermore figure 2 shows the amplitude distribution of the multisines shown in figure 1. It can be seen that due to the homogenization the amplitudes can be scaled up while still satisfying the boundary conditions of $\max |u_k| \leq 1$. This

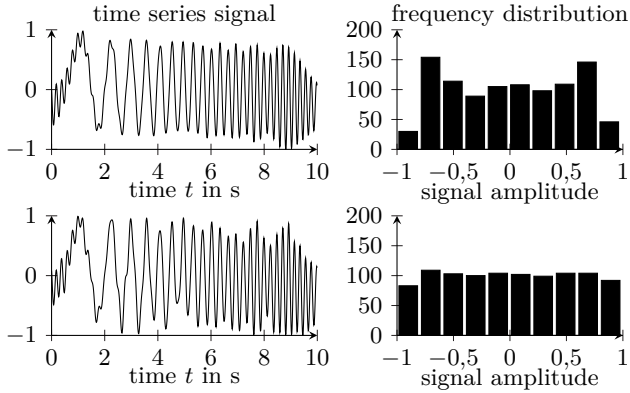


Figure 1: Examples of Multisine signals with Schroeder phases (top) and after homogenization (bottom)

allows a higher energy transfer while keeping the shape of the amplitude spectrum, the excited frequencies and the experiment time as it was in the initial signal. If Schroeder phases are used, the gain in amplitude level is very small, because Schroeder phases already have a good crest factor (Schroeder Phases: $c_R = 1.89$, after Homogenization: $c_R = 1.77$) Also the initialization with Schroeder phases leads to a homogenized signal that closely resembles the initial signal so that it can be assumed that the phase configuration is in the region of a local optimum.

3.2.2 APRBS signals

To generate an APRBS-Signal a shift register is used [7]. At first a Pseudo-Random Binary Sequence is generated. This sequence depends on the structure of the given shift register and the used initialization. To obtain the multilevel signal from this sequence a change of the binary state is used to determine the change of the signal levels. The number of following equal states determines how long a level is held. By considering the holding time T_H , the sampling time T_S and the choice of the amplitude levels, the signal can be generated. For example a pseudo-random binary sequence may be given as:

$$\text{sPRBS} = [1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0] \quad (15)$$

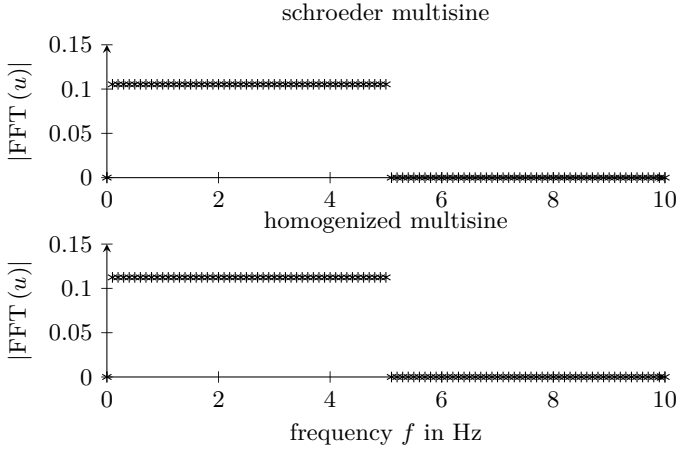


Figure 2: Impact of the homogenization (frequency domain)

The sequence \mathbf{s}_{PRBS} (15) is generated from a shift register with $n = 3$ states and therefore the repeating sequence has 2^{n-1} elements. Consecutive equal states are contracted to indicate a single held amplitude with a weight that will be proportional to the length of this amplitude. In this example, the first amplitude will be held for one T_{H} , the second one for two T_{H} and so on.

$$\mathbf{S} = [1 \quad 2 \quad 3 \quad 1] \quad (16)$$

The length of \mathbf{S} determines the number of amplitudes A_i needed to generate a real signal. With a holding time T_{H} , the sampling time T_{S} and the amplitudes A_i the timeseries is constructed by repeating the amplitudes $A_i \cdot S(i) \cdot \frac{T_{\text{H}}}{T_{\text{S}}}$ times.

The amplitudes can be chosen in different ways. Since there is no model free method available to design the amplitudes that generally yields good results, most of the time the amplitudes are chosen to be either normally or uniformly distributed or according to a sobol-sequence. There is also the possibility of optimizing the crest factor with respect to the amplitude levels but there have been no significant improvements compared to the randomly chosen amplitudes. In the case study presented at the end of this contribution, normally and uniformly distributed amplitudes have been selected. Examples will be given in section 4.

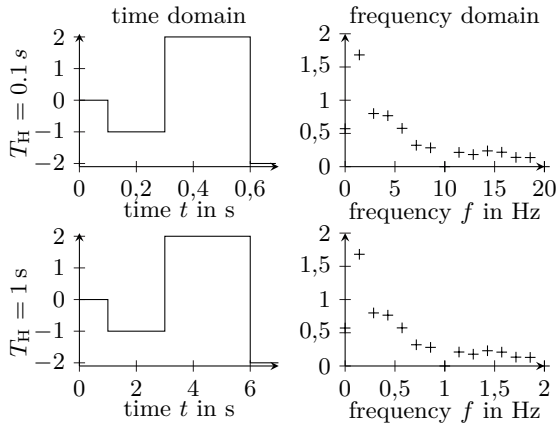


Figure 3: Impact of the holding time

Figure 3 shows the difference between APRBS signals with identical amplitudes but different holding times, which has been exaggerated to show the effect in the frequency domain. If a shorter holding time is used, higher frequencies are excited. Since the amplitude levels and ratios between the lengths of held levels are identical for both inspected holding times, the amplitude spectra can be compared. Figure 3 shows that the shape of both spectra is identical. It is only stretched by the same factor that relates the holding times. A longer holding time results in longer signals and therefore in more data points for identical sampling times.

3.3 Comparison of multisine signals and APRBS

Table 1 provides for a comparison of the two signal types with respect to the signal properties.

3.4 Assessment Criteria

The assessment criteria are divided into two groups. On the one hand test signal properties are analyzed. These include the coverage in the time domain, the crest factor and the signal energy. On the other hand it is of great interest whether the used signals can be used to obtain suitable

identification data. Therefore two measures of the prediction error are used. The RMSE is used as an averaging measure as well as the MAE as a localized measure. It is also interesting to see, how the signal properties relate to suitable identification data.

It is obvious that the crest factor is lower for a signal that is distributed more uniformly. A measure proportional to the energy of a signal x_k is calculated in the frequency domain from the fourier transform $U_k = \text{FFT}(u_k)$:

$$E = \sum_{k=1}^N U_k^2 \tag{17}$$

For a measured signal y_k and its prediction \hat{y}_k of length N the *root mean squared error* (RMSE) is calculated as follows:

$$J_{\text{RMSE}} = \sqrt{\frac{1}{N} \sum_{k=1}^N (y_k - \hat{y}_k)^2} \tag{18}$$

The *maximum absolute error* (MAE) is defined as:

$$J_{\text{MAE}} = \max_{1 \leq k \leq N} |y_k - \hat{y}_k| \tag{19}$$

Table 1: Comparison of multisine signals and APRBS

Signal	Crest Factor	Adjustability	Field of Use
Multisine	The crest factor typically is $c_R = 1.7$ when all amplitudes are chosen to be identical. In general the crest factor of a multisine depends on the choice of amplitudes and phases	The amplitude spectrum can be designed freely and with sufficient excited frequencies the time domain signal can be adjusted while keeping the spectrum constant.	If allowed, multisines are a good approach. If used with random phases the time domain signal resembles random noise, if used with Schroeder or optimized phases the signal range is used more efficiently.
APRBS	For a PRBS signal the crest factor typically is $c_R = 1$ if all power is considered. For an APRBS signal the crest factor depends on the choice of the amplitude levels.	It is harder to adjust the properties of an APRBS. The holding time can be used to influence the spectrum without changing the shape of the spectrum.	The choice of a multilevel signal often depends on plant friendliness. If the holding time is long with respect to system dynamics, static properties can be identified.

The definitions of (18) and (19) are taken from [8]. Since the model approach is a superposition of affine local models the nonlinearity is modelled through the partitioning.

4 Case study

4.1 Test system

The presented case study is based on a theoretical system first introduced in [9] which also has been recorded as a benchmarking example in [8]. The system is given as a nonlinear second order order difference equation:

$$y(k) = \frac{y(k-1) \cdot y(k-2) \cdot (y(k-1) + 2.5)}{1 + (y(k-1))^2 + (y(k-2))^2} + u(k-1) \quad (20)$$

The system has two equilibrium points in $y(k) = 0$ and $y(k) = 2$ which are asymptotically stable in a sufficiently large neighborhood which is not shown here. During the case study the system will be operated around the $y(k) = 0$ equilibrium point. Therefore one obtains:

$$\begin{aligned} y(1) &= 0 \\ y(2) &= 0 \end{aligned} \quad (21)$$

To emulate measurement noise, the output has been disturbed with an independent, identically and normally distributed sequence of random variables:

$$e(k) = \mathcal{N}(0, 0.2^2) \quad (22)$$

The measured system output then results in:

$$y_{\text{meas}}(k) = y(k) + e(k) \quad (23)$$

4.2 Excitation signals

Various APRBS and multisine signals have been used to excite the system. For the multisine signal three different experiment durations have been chosen: $T_1 = 25$ s, $T_2 = 50$ s and $T_3 = 100$ s as well as three maximum frequencies $f_1 = 5$ Hz, $f_2 = 10$ Hz and $f_3 = 100$ Hz. For each of these

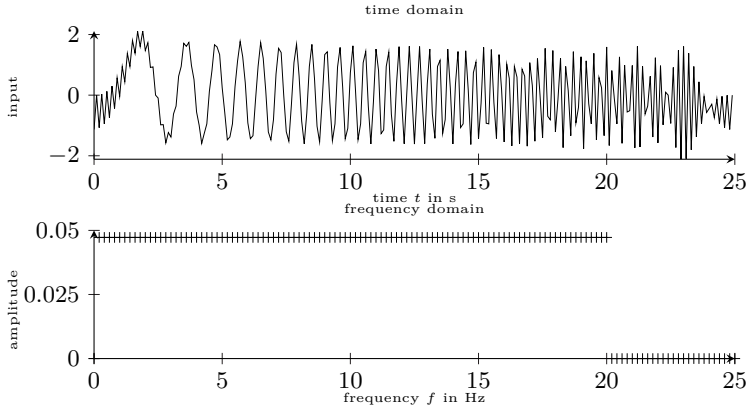


Figure 4: Exemplary multisine identification signal (corresponding identification in figure 7)

combinations there are initializations with Schroeder phases and the respective homogenized signal. For each excitation signal 10 simulations with different noise realizations have been done. This leads to 180 identifications for multisine signals. For the APRBS signals three different seeds of length 8 and three holding times $T_{H,1} = 0.1$ s, $T_{H,2} = 0.2$ s and $T_{H,3} = 0.4$ s have been used. The amplitudes have either been chosen to be normally or uniformly distributed. With 10 simulations for different noise realizations, the number of APRBS identification matches the number of multisine identification at 180. The input signals have been rescaled in a way that a range of $u_{\text{input}} \in]-2, 2[$ is utilized fully. Figures 4 and 5 show exemplary APRBS and multisine identification signals and figure 6 shows the used validation signals which is a randomly generated ramp step function where the amplitudes are held for identical times and the ramp's slopes depend on the time. All 360 models are validated on this data set.

In the case of multisine signals it is the upper frequency bound (f) and in the case of the APRBS signal it is the seed (s) that is used to obtain a pseudo random binary sequence from the shift register. The duration (d) in seconds of the APRBS directly corresponds with the Holding time T_H and have been chosen in a way to match the duration of the multisine signals. The modifications (m) of the multisine signals are the original or initial multisine signals generated with Schroeder phases (org) and the homogenized ones (hom) that lead to a more uniform distribution in the

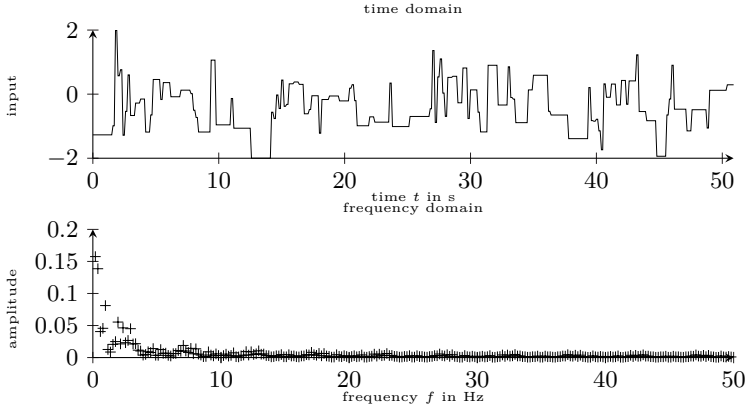


Figure 5: Exemplary APRBS identification signal (corresponding identification in figure 8)

time domain as mentioned. The (corresponding) modifications (m) for the APRBS signals are the choice of amplitudes. They are either distributed normally (n.) or uniformly (u.) where the uniform distribution leads to a more uniform distribution of the signal in the time domain when compared to the normally distributed. Since this modification has a similar effect on the the signal it is assumed that it is allowed to compare the signals in this regard.

4.3 Identification approach and model evaluation

The number of local models $c = 6$ has been determined by cluster validation measures like in [3]. All local models are chosen to have the same structure. For simplicity, the signal terms appearing in 20 have also been used for the model, i.e. $n_u = 1$ and $n_y = 2$. Therefore the regression vector is:

$$\boldsymbol{\varphi}^T(k-1) = [u_{k-1} \quad y_{k-1} \quad y_{k-2}] \quad (24)$$

Since at this point there is no further knowledge available on how to choose the scheduling variable, the first assumption is to choose it identically to the regression vector. Therefore one gets:

$$\mathbf{z}(k-1) = \boldsymbol{\varphi}^T(k-1) = [u_{k-1} \quad y_{k-1} \quad y_{k-2}] \quad (25)$$

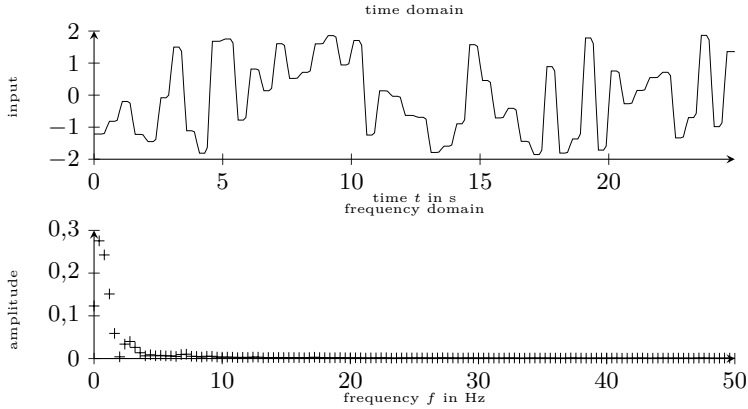


Figure 6: validation signal (corresponding model evaluations in figures 9 and 10)

The fuzziness parameter for the clustering, estimation and model evaluation have been chosen to be identical. To guarantee sufficiently smooth model transitions it was set to $\nu_{\text{clust}} = \nu_{\text{est}} = \nu_{\text{eval}} = \nu = 1.3$. The model's quality is assessed in NOE-configuration. Since the maximum lag is $n_{\text{max}} = 2$ the simulations are initialized with the first two values of the simulated signal.

4.4 Identification results

Due to the homogenization according to the presented method it was possible to achieve a more uniformly covered scheduling space since the system's input is part of that space. Furthermore it was possible to achieve a higher energy transfer to the system since the time series was scaled up to reach higher amplitudes at the excited frequencies with respect to the multisines with Schroeder phases. In this section the complete identification and validation data will be presented in tabular form in table 2. Four exemplary model evaluations including residuals and their distributions have been selected to illustrate the results (see figures 7-10). The shown model evaluations are two identification data sets (one multisine and one APRBS identification (figures 4 and 5)) and the two corresponding evaluations on the validation set. Since the modification parameters for APRBS (choice of amplitudes and seed) and multisine signals (upper frequency and time

domain coverage) do not completely correspond the remaining parameter is in the last column that determines the input signal used for identification.

The RMSE values are given with their mean value and standard deviation for ten repetitions of each experiment. ‘*i*’ denotes identification data, ‘*v*’ denotes validation data, ‘*osap*’ stands for the linear least squares estimation of the local model parameters and ‘*opt*’ stands for the nonlinearly optimized parameters.

Figure 7 and 8 show the model evaluations on the identification data. As expected, the nonlinear optimization procedure results in a better prediction. The multisine chosen for the identification (figure 7) shown in figure 4 is marked with bold font in the first half of table 2 The APRBS chosen for the identification (figure 8) shown in figure 5 is marked with bold font in second half of table 2. Other meaningful values are also marked with bold font in both tables.

From figure 7 it can be seen that the model evaluation matches the simulated system’s output very well. From table 2 it is concluded that the homogenized signals in general yield in better identification results. Since Schroeder phases already yield an acceptable distribution the effect of the homogenization is very small but noticeable. It can be seen in figure 8 that there is a bias on the APRBS identification data when evaluated with *osap* parameters. This bias is reduced by the nonlinear optimization. This effect can be observed for all APRBS identifications. The choice of the amplitudes impact the model quality as well as can be seen in table 2. On the validation data the optimization yields similar results in that the bias is reduced. The most interesting thing is to see whether the (optimized) model performs well on the validation data. For the APRBS identifications it can be stated that the prediction quality of the model with respect to the used validation data is increased by up to 100% due to the optimization. The spectra seen in figures 5 and 6 look similar which apart from the similarities in the time domain indicate a similar excitation which is beneficial for identification. There are lower values for the normally distributed APRBS signals but the ones with a uniform distribution seem to be more consistent and are lower on average. The impact of the seed cannot be isolated at this point.

4.5 Discussion

In this contribution the amplitudes (levels) of the APRBS signals have been chosen randomly (in two different ways) and it was observed that the choice

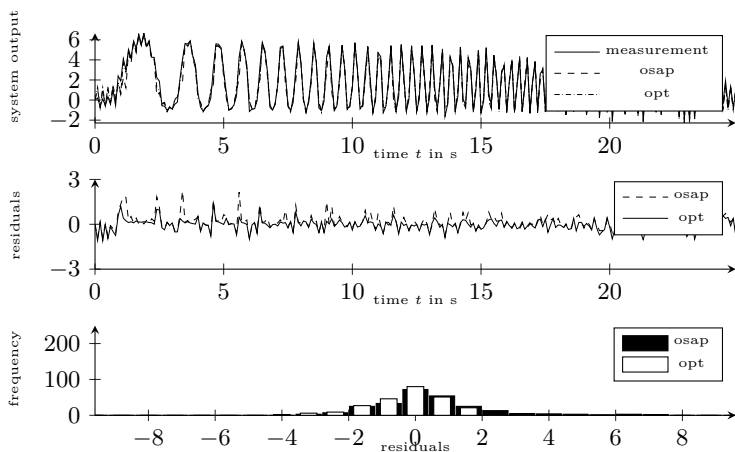


Figure 7: Exemplary identification on a multisine dataset

of amplitudes is significant. Since the homogenization and the choice of uniformly distributed amplitude levels for the APRBS signals have a similar effect on the signal it is concluded that the similarities seen in table 2 are influenced by this macroscopic signal property. The homogenized multisine as well as the APRBS signal with uniformly distributed amplitudes yield worse identification results compared to the respective timeseries without uniform distribution. Therefore it is concluded that the distribution of the timeseries is significant for the identification results, although at this point no rule for the choice of the distribution can be given. In [10] a different case study has been investigated. From this case study it could also be concluded that the time domain distribution is significant. In that case a more uniform distribution was beneficial to the identification results. The test signal design, identification and evaluation of the case study in [10] has been conducted in the same way.

While measures like energy transfer can be used to design and choose a test signal before taking any knowledge about the system under consideration, the amplitudes will have to be used to fit the test signal to the specific process that is to be identified. From this contribution and [10] the significance of the choice of amplitudes is validated.

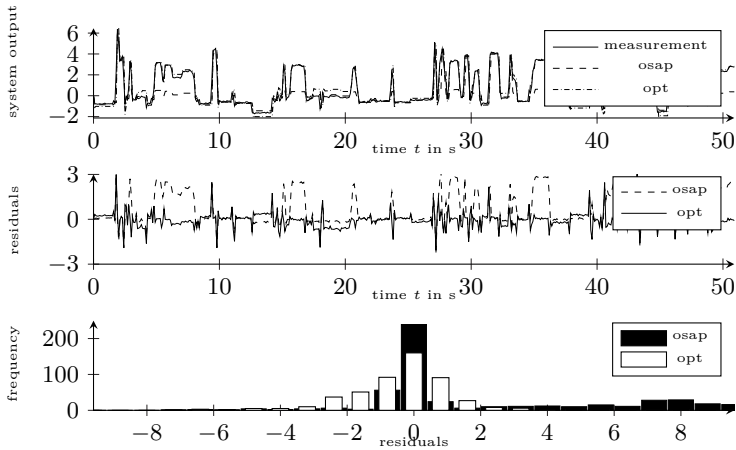


Figure 8: Exemplary identification on an APRBS dataset

5 Conclusion and outlook

The parametrization of standard test signals has been investigated. It is concluded that the degrees of freedom that are not used during the design process play a central role when it comes to the quality of the identified model. Design choices that are not taken consciously will be taken automatically which is an undesirable situation for a designer. In this paper two major types of standard test signals have been investigated. There can be requirements for the designer that prohibits the usage of harmonic excitations, therefore it is important to be able to parametrize step based signal like the APRBS to achieve a high quality model. A usual process while using standard test signals is to only adhere to the boundaries of the actuators without trying to match the frequency band of operation. One also should use a signal similar to the system's input during operation. Since the inserted energy depends only on the amplitudes and not on the holding time, it can be used to target the frequency band of interest after the desired amplitude distribution has been achieved.

Since the properties in the frequency domain of chirp signals are very similar to multisines but harder to design properly they have not been considered in this work. Future research will also investigate the way to choose the amplitudes of both multisines and aprbs signals.

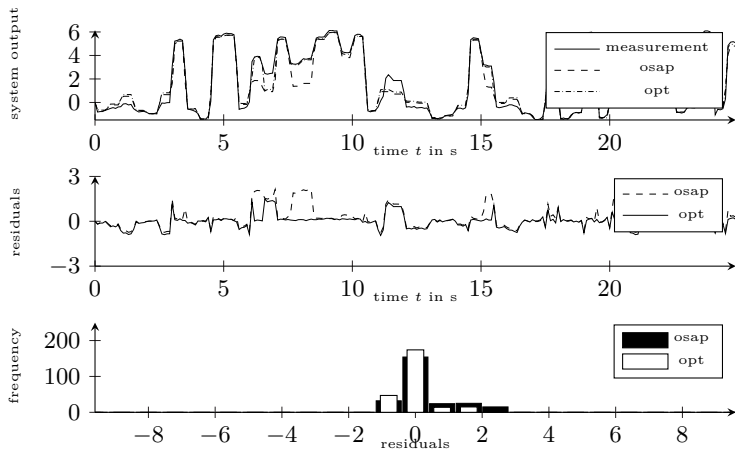


Figure 9: Exemplary validation based on a multisine identification

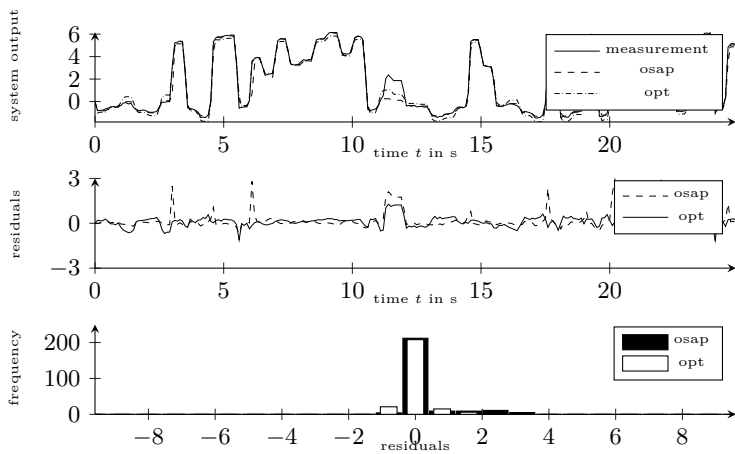


Figure 10: Exemplary validation based on an APRBS identification

Table 2: RMSE values for multisine and APRBS identifications (o: schroeder phases, h: homogenized, n: normally distributed amplitudes, u: uniformly distributed amplitudes, f: upper frequency, s: seed, d: duration, i: identification data, v: validation data)

m	d	f	E	i-osap	i-opt	v-osap	v-opt
o	25	5	1.1	0.53 ± 0.004	0.39 ± 0.001	0.81 ± 0.003	0.49 ± 0.001
o	25	10	1.12	0.55 ± 0.002	0.43 ± 0.001	0.98 ± 0.001	0.63 ± 0.002
o	25	20	1.31	0.96 ± 0.002	0.69 ± 0.001	1.33 ± 0.001	1.02 ± 0.005
o	50	5	1.1	0.5 ± 0.002	0.37 ± 0.001	0.8 ± 0.003	0.5
o	50	10	1.12	0.55 ± 0.002	0.43 ± 0.001	0.97 ± 0.001	0.63 ± 0.001
o	50	20	1.32	0.96 ± 0.002	0.69 ± 0.001	1.34 ± 0.001	1.09 ± 0.001
o	100	5	1.1	0.5 ± 0.001	0.38 ± 0.001	0.8 ± 0.003	0.49 ± 0.001
o	100	10	1.11	0.54 ± 0.001	0.42 ± 0.001	0.96 ± 0.001	0.63 ± 0.001
o	100	20	1.32	0.94 ± 0.001	0.69 ± 0.001	1.32 ± 0.001	1.09 ± 0.001
h	25	5	1.1	0.52 ± 0.004	0.4 ± 0.001	0.86 ± 0.003	0.52
h	25	10	1.3	0.57 ± 0.002	0.45 ± 0.001	1.02 ± 0.006	0.67 ± 0.002
h	25	20	1.31	0.96 ± 0.002	0.74 ± 0.001	1.37 ± 0.001	1.13 ± 0.002
h	50	5	1.12	0.5 ± 0.003	0.38 ± 0.001	0.82 ± 0.004	0.52
h	50	10	1.31	0.57 ± 0.001	0.45 ± 0.001	1 ± 0.001	0.68 ± 0.001
h	50	20	1.32	0.98 ± 0.002	0.75 ± 0.001	1.37 ± 0.001	1.14 ± 0.002
h	100	5	1.12	0.5 ± 0.001	0.38 ± 0.001	0.82 ± 0.001	0.5
h	100	10	1.32	0.56 ± 0.001	0.43 ± 0.001	0.99 ± 0.001	0.67 ± 0.002
h	100	20	1.33	0.97 ± 0.001	0.88 ± 0.001	1.36 ± 0.001	1.51 ± 0.034
m	d	s	E	i-osap	i-opt	v-osap	v-opt
n	25	1	0.65	1.43 ± 0.01	0.55 ± 0.001	1.02 ± 0.003	0.58 ± 0.01
n	25	2	0.58	0.9 ± 0.008	0.83 ± 0.004	0.68 ± 0.007	1.85 ± 0.01
n	25	3	1.43	1.03 ± 0.011	0.54 ± 0.001	0.83 ± 0.005	0.62 ± 0.015
n	50	1	0.73	1.15 ± 0.001	0.36 ± 0.021	0.8 ± 0.003	0.37 ± 0.005
n	50	2	0.52	1.25 ± 0.002	0.48 ± 0.17	1.11 ± 0.001	0.46 ± 0.18
n	50	3	1.36	0.84 ± 0.001	0.5 ± 0.001	1.2 ± 0.002	0.46
n	100	1	0.67	1.65 ± 0.006	0.47	1.12 ± 0.002	0.4 ± 0.001
n	100	2	1.03	1.24 ± 0.004	0.53 ± 0.005	1.09 ± 0.002	0.57 ± 0.045
n	100	3	1.49	1.12 ± 0.001	0.39	1.22 ± 0.003	0.4 ± 0.001
u	25	1	0.69	0.91 ± 0.004	0.58	0.96 ± 0.003	0.47 ± 0.001
u	25	2	1.49	1.38 ± 0.07	0.65 ± 0.2	1.03 ± 0.003	0.5 ± 0.2
u	25	3	1.2	1.05 ± 0.003	0.51 ± 0.001	0.94 ± 0.008	0.41
u	50	1	0.79	1.38 ± 0.02	0.57 ± 0.001	1.12 ± 0.001	0.42 ± 0.002
u	50	2	1.28	1.07 ± 0.001	0.54 ± 0.001	1.14 ± 0.003	0.43
u	50	3	1.42	1.24 ± 0.002	0.49 ± 0.001	1.11 ± 0.001	0.42
u	100	1	0.68	0.94 ± 0.002	0.43 ± 0.001	0.98 ± 0.003	0.43
u	100	2	1.46	1.23 ± 0.002	0.37 ± 0.01	1.12 ± 0.001	0.49 ± 0.03
u	100	3	1.46	1.15 ± 0.002	0.46 ± 0.01	1.09 ± 0.001	0.44 ± 0.002

References

- [1] C. Hametner, M. Stadlbauer, M. Deregnaucourt, M. Jakubek, T. Winsel, “Optimal experiment design based on local model networks and multi-player perceptron networks,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 251–261, 2013.
- [2] A. Kroll, A. Dürrbaum, “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)*, Beijing, China, October 19–21, pp. 1427–1432, 2015.
- [3] M. Gringard, A. Kroll, “On the homogenization of broadband test signals for nonlinear system identification and the application to an electro-mechanical actuator [in German],” in *Proc. 25th Workshop Computational Intelligence*, Dortmund, Germany, November 26–27, pp. 145–162, 2015.
- [4] A. Kroll, “On choosing the fuzziness parameter for identifying TS models with multidimensional membership functions,” *Journal of Artificial Intelligence and Soft Computing Research*, vol. 1, no. 4, pp. 283–300, 2011.
- [5] S. Duym, J. Schoukens, “Design of excitation signals for the restoring force surface method,” *Mechanical Systems and Signal Processing*, vol. 9, no. 2, pp. 139–158, 1995.
- [6] M. Schroeder, “Synthesis of low-peak-factor signals and binary sequences with low autocorrelation,” *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 85–89, 1970.
- [7] K. Godfrey, *Perturbation Signals for System Identification*, 1st edition. Prentice Hall, 1993.
- [8] A. Kroll, H. Schulte, “Benchmark problems for nonlinear system identification and control using Soft Computing methods: Need and overview,” *Applied Soft Computing*, vol. 25, pp. 496–513, 2014.
- [9] K. S. Narendra, K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” *IEEE Transactions on Neural Networks*, vol. 1, no. 1 pp. 4–27, 1990.
- [10] M. Gringard, A. Kroll, “On the parametrization of APRBS and multisine test signals for the identification of nonlinear dynamic TS-models,” in *Proc. 7th IEEE Symposium Series on Computational Intelligence*, Athens, Greece, December 6–9, 2016.

Vergleich von Anregungssignalen für Nichtlineare Identifikationsaufgaben

Tim Oliver Heinz, Oliver Nelles

Universität Siegen
Department Maschinenbau
Paul-Bonatz-Str. 9-11
57068 Siegen
E-Mail: tim.heinz@uni-siegen.de

Kurzfassung

Aufgrund wachsender Dynamikanforderungen im Bereich der Regelungs- und Steuerungstechnik gewinnt die Identifikation nichtlinearer dynamischer Black-Box-Modelle zunehmend an Aufmerksamkeit. Bei solchen datenbasierten Modellierungsansätzen hängt die Modellqualität direkt von dem verwendeten Datensatz (Ein- und Ausgangssignale) ab. Bei der Identifikation in NARX-Konfiguration setzt sich der Eingangsraum des Identifikationsproblems aus verzögerten Ein- und Ausgängen des Prozesses zusammen. Eine gleichmäßige Abdeckung dieses Eingangsraumes ist vorteilhaft für die nichtlineare Identifikation. Durch Vorgabe eines Anregungssignals können jedoch nur verzögerten *Eingänge* des Prozesses direkt beeinflusst werden. Das daraus resultierende Ausgangssignal kann nur indirekt beeinflusst werden. Dies erschwert die Aufgabe, eine gewünschte Punktverteilung im Eingangsraum zu erzeugen, erheblich.

Bekannte Eingangssignale für die Identifikation von dynamischen Modellen sind Chirp, Multisinus, Zufallssignale, Rampen, sowie pseudo zufällige binäre Signale (PRBS) und dessen amplitudenmodulierte Erweiterung (APRBS). Diese Signale weisen stark unterschiedliche Eigenschaften (Frequenzabdeckung, Amplitudenabdeckung, Eingangsraumabdeckung, ...) auf, sodass sich nur wenige hiervon für nichtlineare Identifikationsaufgaben eignen. Die Wahl des Eingangssignals für die Vermessung nichtlinearer dynamischer Prozesse spaltet die wissenschaftliche Welt in zwei Lager: Auf der einen Seite stehen die Sinus-basierten Signale (Chirp, Multisinus), mit denen sich ein gewünschtes Frequenzspektrum einstellen lässt. Im Kontrast hierzu stehen Sprung-basierte Signale (bspw. APRBS), welche verschiedene

Betriebspunkte durch schnelle Änderungen anfahren. Beide Signaltypen werden in der Literatur für Identifikationsaufgaben genutzt [16, 15, 7]. Eine grundlegende Gegenüberstellung fehlt bislang.

Ziel dieser Veröffentlichung ist der ausführliche Vergleich der häufig verwendeten Signaltypen Chirp, Multisinus und APRBS. Hierbei werden die Signale mit Hinblick auf Approximationsqualität, Extrapolationssicherheit und Eingangsraumabdeckung mit Hilfe von nichtlinearen synthetischen Prozessen untersucht. Es zeigt sich einerseits, dass Sprung-basierte Signale den Eingangsraum sehr gleichmäßig abdecken, jedoch sehr hohe Verfahrgeschwindigkeiten in den Übergangsbereichen aufweisen. Andererseits sind die Gradienten von Sinus-basierten Signale deutlich geringer und können auf das gewünschte Frequenzband richtungsabhängig abgestimmt werden. Im Hinblick auf die Eingangsraumabdeckung scheidet der Sinus-basierte Signaltyp deutlich schlechter ab. Für die Identifikation von nichtlinearen Prozessen in der Praxis ist die Beschränkung der Verfahrgeschwindigkeit ebenso relevant, wie die gleichmäßige Abdeckung des Eingangsraumes.

Diese Erkenntnisse motivieren die Vorstellung eines neuen Signaltyps, welcher die Vorteile beider untersuchter Typen vereint, ohne deren nachteilige Eigenschaften zu übernehmen. Als Basis für den neuen Signaltyp dient ein APRBS. Die unerwünschten sprungförmigen Übergänge werden durch stückweise Sinusfunktionen ersetzt. Die freien Parameter der Sinusfunktionen werden so gewählt, dass ein stetiger Signalverlauf entsteht, welcher die gewünschte Beschränkung der Gradienten einhält. Darüber hinaus ist das Signal stetig differenzierbar und in den Konstanten- und den Übergangsbereichen sogar glatt. Abschließend wird der neue Signaltyp mit den zwei untersuchten Signaltypen verglichen und die Identifikationsqualität beurteilt.

1 Einführung

1.1 Identifikation nichtlinearer dynamischer Systeme

Für die Identifikation nichtlinearer dynamischer Modelle ist die Strategie der ausgelagerten Dynamik (external dynamics) üblich (vgl. [6, 5]). Hierbei kann das nichtlineare Modell klar in einen nichtlinearen statischen Approximator und eine Filterbank für die Dynamik aufgeteilt werden. Aufgrund der Vielfalt verschiedener Dynamikrealisierung für die Filterbank wird sich hier

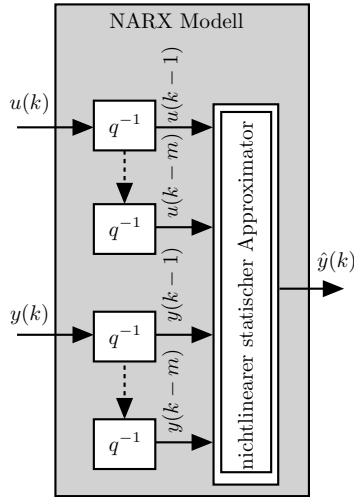


Bild 1: Modellstruktur NARX: Die Filterbank setzt sich aus Zeitverschiebungsgliedern (q^{-1}) für Ein- und Ausgang zusammen.

auf die häufig verwendete NARX-Struktur konzentriert, welche schematisch in Bild 1 gezeigt ist.

Aus Bild 1 wird ersichtlich, dass die Eingänge des nichtlinearen statischen Approximators sowohl aus verzögerten Prozessein- und -ausgängen bestehen. Die Ausgänge des Prozesses und dessen verzögerte Versionen, welche für die Identifikation benötigt werden, sind nur indirekt durch das Eingangssignal beeinflussbar. Eine gleichmäßige Punktabdeckung im Eingangsraum, welcher unter anderem aus verzögerten Prozessausgängen besteht, wird durch diese Problematik erschwert.

1.2 Anforderungen an Signale

Um einen fairen Vergleich zwischen verschiedenen Signaltypen zu gewährleisten, ist es notwendig Anforderungen an die Signale zu definieren, welche sich einerseits aus dem Bereich der Vermessung von Prozessen, andererseits aus dem Bereich der Identifikation ergeben:

- Beschränkungen der Verfahrensgeschwindigkeit:
In realen Prozessen ist die zulässige Änderungsrate eines Stellgliedes meist begrenzt, um Schädigungen vorzubeugen.
- Möglichkeit zur Einstellung des Frequenzspektrums:
Bei der Annahme einer zu geringen Modellordnung in NARX-Struktur, wird bei der Bestimmung der Parameter der Fehler mit dem Frequenzspektrum des Signals gewichtet [6, 5]. Wenn das Spektrum der späteren Anwendung (näherungsweise) bekannt ist, sollte dies bei der Erzeugung des Signals berücksichtigt werden.
- Datenverteilung:
Eine möglichst homogene Abdeckung des gesamten Eingangsraums des statischen nichtlinearen Approximators durch das Eingangssignal ist wünschenswert. Datenlücken können zu stark extrapolierendem Verhalten des identifizierten Modells führen.

Für die Bewertung von Signalen im Rahmen der linearen Identifikation wird häufig der Crest-Faktor des Signals ermittelt. Dieser ergibt sich aus dem Verhältnis von maximaler Amplitude zu der Wurzel des quadratischen Mittelwertes (RMS) des Signals:

$$Cr(u) = \frac{u_{\text{peak}}}{u_{\text{RMS}}}. \quad (1)$$

Hier wird auf die Bewertung anhand des Crest-Faktors verzichtet, da Signale mit einem optimalen Crest Faktor wie bspw. ein PRBS für nichtlineare Identifikationsaufgaben nicht geeignet sind [7]. Im Folgenden werden die bekannten Signaltypen hinsichtlich dieser Anforderungen untersucht.

2 Vergleich bestehender Signaltypen

In diesem Abschnitt sollen bestehende populäre Signaltypen auf die Eigenschaften des vorherigen Abschnitts untersucht werden. Fokus liegt hierbei auf dem Vergleich von *passiven* Signalen für eine anschließende nichtlineare Identifikation, welche ohne tieferes Wissen über den Prozess auskommen. Hierzu wird pro Signal ein Datensatz mit Hilfe eines nichtlinearen dynamischen Prozesses erzeugt. Die Signale haben eine Länge von $N = 1000$ Datenpunkten und decken die Amplitudenwerte von $u(k) \in \{-1, \dots, 1\}$ ab.

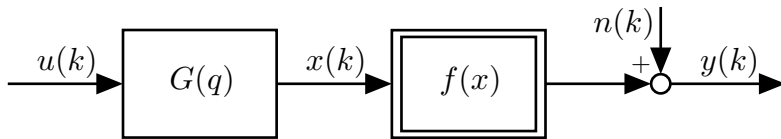


Bild 2: Blockschaltbild eines nichtlinearen Systems mit Wiener-Struktur.

Für den Vergleich der verschiedenen Signaltypen im Eingangsraum des Identifikationsproblems wird ein nichtlinearer Prozess in Wiener-Struktur verwendet (siehe Bild 2). Die Nichtlinearität hat Sättigungscharakter und wird durch einen Arkustangens dargestellt: $f(x) = \frac{1}{1.33} \arctan(4x)$. Aus Gründen der Darstellung wird ein System erster Ordnung für die lineare Dynamik angenommen: $G(q) = \frac{0.1q^{-1}}{1-0.9q^{-1}}$. Somit wird der Eingangsraum durch einen verzögerten Eingang und einen verzögerten Ausgang aufgespannt: $u(k-1)$ und $y(k-1)$. Der Einfluss von Rauschen am Prozessausgang wird nicht berücksichtigt ($n(k) = 0$), sodass mögliche stochastische Einflüsse ausgeschlossen sind.

2.1 Sinus-basierte Signale

Eine wichtige Gruppe von Signaltypen basiert auf Sinusfunktionen. Die wichtigsten Stellvertreter sind Chirp- und Multisinus-Signale. Diese werden im folgenden kurz vorgestellt und typische Anwendungen in der Literatur aufgezeigt.

2.1.1 Chirp

Ein Chirp-Signal (oder auch *swept sine*) basiert auf einer Sinusfunktion mit variabler Frequenz. Allgemein kann diese Funktion durch

$$u(t) = A \sin(f(t) + \varphi_0) \quad (2)$$

beschrieben werden. Hierbei ist $f(t)$ meist eine monotone Funktion und wird häufig als linear oder affin angenommen. Der Wert A beschreibt die Amplitude des Signals, die optionale Phasenverschiebung φ_0 erlaubt es Anfangsbedingungen einzustellen.

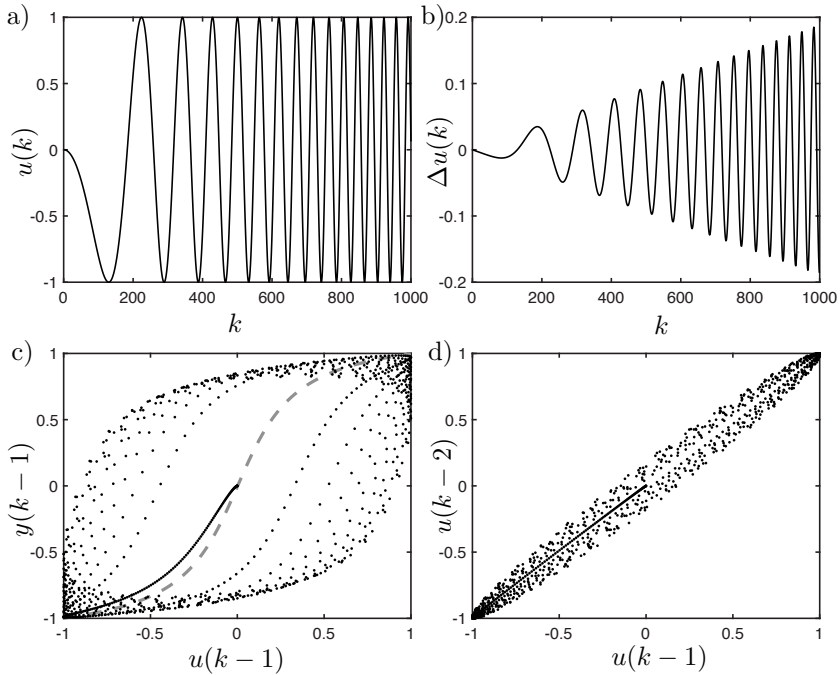


Bild 3: a) Signalverlauf eines Chirp-Signals.
 b) Zeitliche Ableitung.
 c) Eingangsraum für ein NARX Modell 1. Ord., Statik grau gestrichelt.
 d) Eingangsraum bestehend aus verzögerten Prozess Eingängen.

Die Verwendung von Chirp-Signalen gewinnt an Popularität und Bedeutung. Für die Anwendung im Bereich der Verbrennungsmotorentechnik wurde diese Signal in [1] vorgeschlagen. Eine Erweiterung des Signaltyps für die Anwendung mit Beschränkungen für das Eingangssignal findet sich in [16].

Für den Vergleich wurde ein Chirp-Signal erzeugt, welches die Frequenz linear ansteigen lässt und in den Bereich von $f_{\min} = 10^{-5}$ bis $f_{\max} = 3 \cdot 10^{-1}$ legt, sodass eine maximaler Gradient des Signals von $|\dot{u}_{\max}| = 0.25$ nicht überschritten wird. Der Signalverlauf, dessen zeitliche Ableitung und die Punkteverteilung im Eingangsraum sind in Bild 3 gezeigt.

Die Abdeckung des Eingangsraums mit Datenpunkten kann qualitativ anhand des Bildes erfolgen. In c) ist eine deutliche Datenhäufung im

Bereich minimaler und maximaler $u(k-1)$ zu sehen. Im Gegensatz hierzu steht (i) der dünn besetzte mittlere Bereich und (ii) die unbesetzte linke obere und rechte untere Ecke. Eine solche Punktverteilung ist die direkte Folge der Charakteristik des Chirp-Signals: Schnell bei mittleren Werten ≈ 0 , langsam in der Nähe der Extrema und ein limitierter Gradient.

Wird dieser Datensatz im folgenden für eine Identifikation genutzt, besteht die Gefahr von lokal schlechter Modellgüte in den Bereichen mangelnder Punktdichte. Um dieser schlechten Abdeckung zu entgehen, werden Chirp-Signale mit anderen Signaltypen, wie bspw. Rampen, kombiniert [16]. Eine langsame, rampenförmige Anregung generiert Punkte in dem schwach besetzten Bereich rund um die statische Kennlinie.

In d) zeigt sich der Eingangsraum bestehend aus zwei verzögerten Prozesseingängen. Dieser wird notwendig für die Identifikation eines Systems mindestens 2. Ordnung. Aufgrund des begrenzten Gradienten des Signals ergeben sich hier nur Werte in der Nähe der Diagonalen. Datenpunkte weit von dieser Diagonalen entfernt würde auf einen großen Gradienten im Signal hinweisen.

Die hier verwendete lineare Funktion $f(t)$ für die Frequenz wird bei der Betrachtung der Einhüllenden der ersten Ableitung des Signals in b) erkennbar. Die Anpassung einer gewünschten maximalen Verfahrensgeschwindigkeit ist bei einem Chirp-Signal vergleichbar einfach. Die Ableitung des Signals ergibt sich zu

$$\frac{du(t)}{dt} = A \frac{df(t)}{dt} \cos(f(t) + \varphi_0). \quad (3)$$

Somit kann die maximale Änderungsrate des Signals durch Anpassung der Amplitude A zusammen mit der Funktion für die Frequenz $f(t)$ eingestellt werden. Die Einstellung des Frequenzspektrums erfolgt ebenfalls über die Funktion $f(t)$. Neben der oben aufgeführten linearen Aufteilung des Frequenzspektrums kann auch eine Betonung von hohen oder niedrigen Frequenzen durch degressive oder progressive Funktionen für $f(t)$ erfolgen.

2.1.2 Multisinus

Ein Multisinus-Signal besteht aus mehreren, überlagerten, monofrequenten Kosinus-Signalen [12]. Die Frequenzen sind hierbei harmonische einer

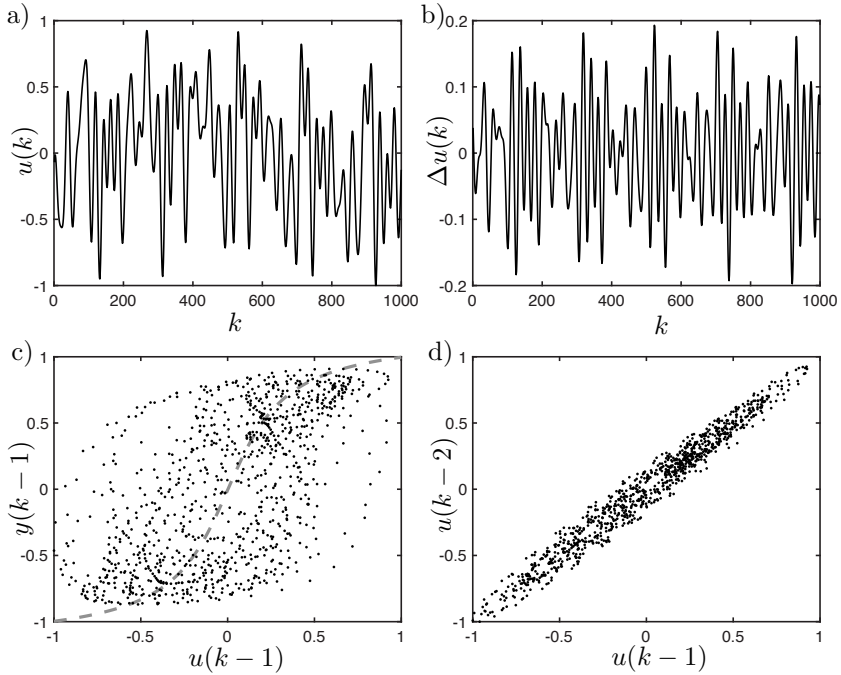


Bild 4: a) Signalverlauf eines Multisinus-Signals.
 b) Zeitliche Ableitung.
 c) Eingangsraum für ein NARX Modell 1. Ord., Statik grau gestrichelt.
 d) Eingangsraum bestehend aus verzögerten Prozess Eingängen.

Grundfrequenz f_0 :

$$u(t) = \sum_{k=1}^F A \cos(2\pi l_k f_0 t + \phi_k) \quad \text{mit } l_k \in \mathbb{N}. \quad (4)$$

Für die Wahl von ϕ_k können die Phasenverschiebungen nach Schroeder verwendet werden [14] um einen möglichst geringen Crest-Faktor zu erreichen. In der Literatur wird auch eine Optimierung der Phasenverschiebung vorgeschlagen.

Multisinus-Signale werden häufig in Frequenz-basierten Identifikationsmethoden (vgl. [9, 13]) eingesetzt. Darüber hinaus werden ebenfalls Ansätze für

die Identifikation von nichtlinearen Zustandsraummodellen verfolgt [8].

Für den Vergleich der Eingangsraumabdeckung wurde ähnlich zum Chirp-Signal der Frequenzbereich $f_{\min} = 10^{-5}$ bis $f_{\max} = 10^{-1}$ gewählt. Hierbei wurden die Phasenverschiebungen ϕ_k durch eine primitive Optimierung gewählt. In Bild 4 ist sowohl der Signalverlauf, dessen zeitliche Ableitung und auch die Eingangsraumverteilung abgebildet.

In c) ist die sehr gleichmäßige Abdeckung mit Datenpunkten des Eingangsraums durch ein Multisinus-Signal zu sehen. Lediglich sämtliche Eckbereiche sind schwach bis gar nicht abgedeckt. Im Gegensatz zum Chirp-Signal ist die Einbringung von Randbedingungen in Form einer maximalen Änderungsrate nicht trivial [11]. In a) ist ein Signal zu sehen, was eine Änderungsrate von $|\dot{u}_{\max}| = 0.2$ nicht überschreitet. Dies ist sowohl in der Ableitung des Signals in b) als auch in der schmalen, bandförmigen Punktverteilung im Eingangsraum der verzögerten u (Bild 4d)) qualitativ zu erkennen.

Das Frequenzspektrum hingegen kann durch die Wahl der Grundfrequenz f_0 und der Anzahl von überlagerten Sinusfunktionen leicht eingestellt werden. Klassischerweise hat jede Sinusfunktion die gleiche Amplitude A , was keine besondere Betonung von einzelnen Frequenzbändern erlaubt.

2.2 Sprung-basierte Signale

Für die Identifikation von linearen Modellen haben sich Sprung-basierte Signale wie ein pseudo zufälliges binäres Signal (PRBS) oder generalisiertes binäres Rauschen (GBN) [17] etabliert. Die stückweise konstanten Signale ermöglichen dem Prozess, in die Nähe des Ruhezustandes zu kommen. Dies betont besonders die stationären Eigenschaften des Prozesses und ist nützlich für eine folgende Identifikation. Für die nichtlineare Identifikation ist es hingegen wichtig neben einem breiten Frequenzspektrum auch ein breites Amplitudenspektrum abzudecken. Dies ist mit diesen – für lineare Prozesses üblichen – Signaltypen nicht möglich

2.2.1 APRBS

In [7] wurde erstmals ausführlicher diskutiert, dass es nicht möglich ist, mit einem PRBS sämtliche nichtlineare Effekte eines Prozesses zu erfassen, da die zwei abwechselnden Levels eines PRBS nicht alle Betriebspunkte treffen.

Wird für jeden Sprung ein anderer Amplitudenwert verwendet, können beliebige stationäre Werte erreicht werden. Diese amplitudenmodulierte Erweiterung (APRBS) wird hier näher betrachtet. Aufgrund verschiedener Prozessdynamiken muss eine minimale Haltedauer T_H für das APRBS vorgegeben werden, um auch bei langsamen Prozessen den stationären Betriebspunkt zu erreichen.

Aufgrund der Eigenschaft, dass ein APRBS viele verschiedene Betriebspunkte anfährt und dabei mitunter auch eingeschwungene Zustände erfasst findet es in vielen Bereichen Anwendung [10, 3].

In Bild 5a) ist ein möglicher Verlauf eines APRBS mit einer minimalen Haltedauer von $T_H = 8$ Zeitschritten zu sehen. Durch die stückweise konstanten Signale ergeben sich in c) vertikale Punktverläufe, welche sich in der Nähe der statische Kennlinie des Prozesses häufen. Der Eingangsraum in c) ist in fast allen Bereich gut mit Datenpunkten bedeckt. Lediglich einige schwach besetzte Bereiche ergeben sich aufgrund der zufälligen Reihenfolge der Sprunghöhen.

In b) ist die Ableitung des Signals zu sehen. Bei jedem Sprung ergeben sich hier sehr große Änderungsraten des Signals. Der Unterraum für die verzögerten Prozesseingänge ist in d) zu sehen. Für die Identifikation von Modellen höherer Ordnung muss auch dieser Raum mit Daten gefüllt sein. Außerhalb der Diagonalen ergeben sich nur vereinzelt Punkte wenn das Signal springt. Man beachte: Die Datenpunkte auf der Diagonale sind aufgrund der minimalen Haltedauer T_H mindestens 8-fach besetzt. Weitere Nachteile des APRBS liegen sowohl in der mangelnden Möglichkeit ein gewünschtes Frequenzspektrum einzustellen als auch in den großen Veränderungsraten bei Amplitudensprüngen.

2.2.2 Weitere Sprung-basierte Signale

Im Bereich der aktiven Versuchsplanung von dynamischen Modellen werden häufig stückweise konstante Signale als initiales Design verwendet. Im Anschluss werden einzelne Signalwerte oder Sequenzen hiervon auf Basis von Randbedingungen oder Varianz der ermittelten Parameter optimiert [18, 4].

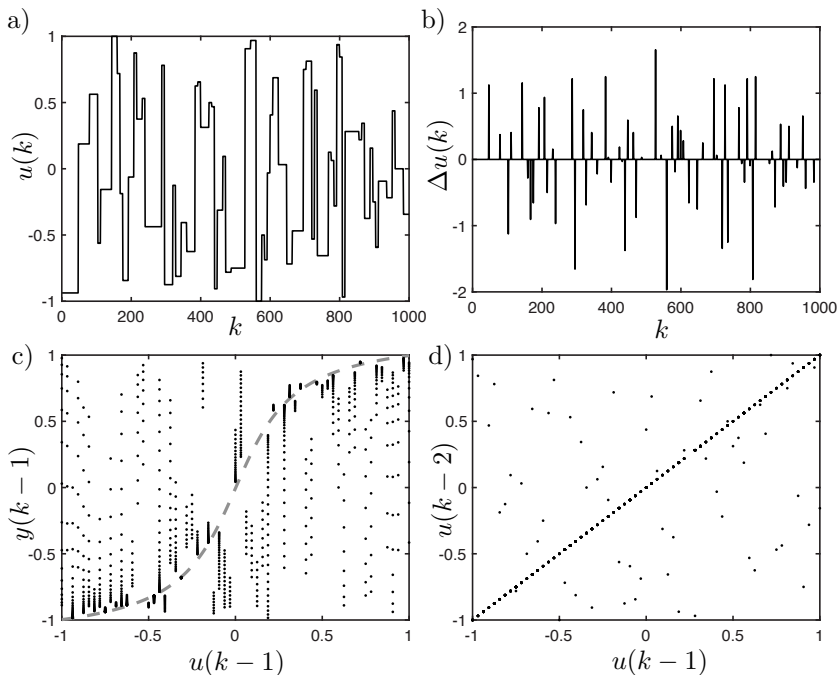


Bild 5: a) Signalverlauf eines APRBS.
 b) Zeitliche Ableitung.
 c) Eingangraum für ein NARX Modell 1. Ord., Statik grau gestrichelt.
 d) Eingangraum bestehend aus verzögerten Prozess Eingängen.

3 Vorschlag eines neuen, kombinierten Signaltyps

Wie im vorherigen Abschnitt erwähnt hat ein APRBS klare Vorteile im Bereich gleichmäßiger Eingangsraumabdeckung. Dies ist auf die stückweise konstanten Signalabschnitte zurückzuführen, welche den Prozess auch nahe des statischen Gleichgewichts bringen. Sinus-basierten Signaltypen haben die Vorteile, einen endlichen Gradienten zu besitzen und die Möglichkeit zur Einstellung eines gewünschten Frequenzspektrums zu bieten.

Im Folgenden wird ein Signaltyp vorgestellt, welcher stückweise aus beiden Signaltypen besteht. Die Basis bildet ein APRBS, welches in den Sprungstellen auf eine maximale Verfahrensgeschwindigkeit herabgesetzt wird. Eine Möglichkeit, diese Randbedingung umzusetzen, besteht in der Anwendung

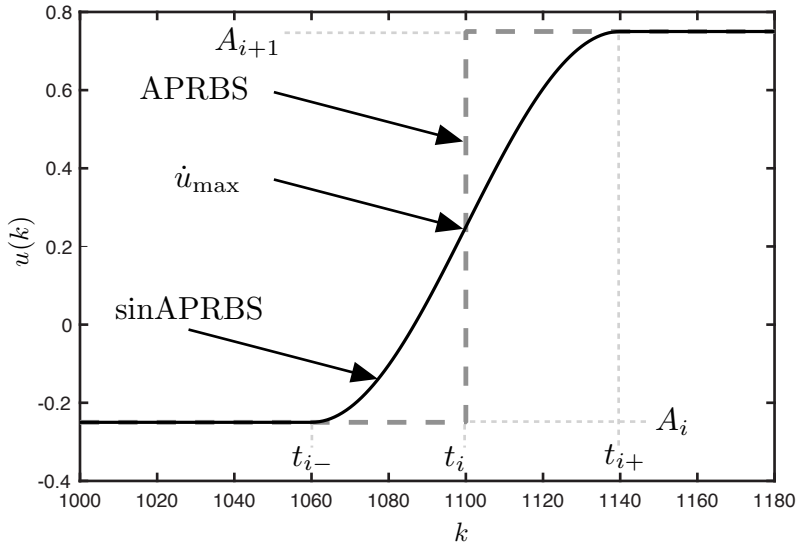


Bild 6: Veranschaulichung eines sinAPRBS.

von Rampen zwischen den Sprungstellen, welches bereits erfolgreich in [3] demonstriert wurde. Um die ungewollten, schnellen Sprünge des Signals zu eliminieren, sollen hier stückweise Sinus-Signale eingesetzt werden. Der Vorteil liegt hierbei in der Differenzierbarkeit des Übergangs zwischen beiden Signalstücken. Dies ergibt eine gute Abdeckung in den Räumen, welche aus mehreren verzögerten Eingängen bestehen. Ein konventionelles APRBS hat in diesen Räumen nur vereinzelt Punkte außerhalb der statischen Diagonalen, wie in Bild 5 zu sehen ist.

In Bild 6 wird die Anpassung eines APRBS Teilstücks durch einen stückweisen Sinusverlauf veranschaulicht. Der funktionale Zusammenhang der Sinus-Übergänge hängt hierbei neben den maximal zulässigen Änderungsraten $|\dot{u}_{\max}|$ von der Differenz der Amplituden A_i und A_{i+1} vor bzw. nach dem Sprung ab:

$$u_{1,i}(t) = \begin{cases} \frac{A_i + A_{i+1}}{2} + \frac{A_{i+1} - A_i}{2} \sin\left(\frac{2|\dot{u}_{\max}|(t - t_i)}{|A_{i+1} - A_i|}\right) & , \quad t \in \{t_{i-}, \dots, t_{i+}\} \\ 0 & , \quad \text{sonst} \end{cases} \quad (5)$$

mit

$$t_{i-} = t_i - \frac{\pi |A_{i+1} - A_i|}{4 |\dot{u}_{\max}|} \quad \text{und} \quad t_{i+} = t_i + \frac{\pi |A_{i+1} - A_i|}{4 |\dot{u}_{\max}|}. \quad (6)$$

Die konstanten Signalabschnitte werden mathematisch wie folgt beschrieben:

$$u_{2,i}(t) = \begin{cases} A_i & , \quad t \in \{t_{(i-1)+}, \dots, t_{i-}\} \\ 0 & , \quad \text{sonst} \end{cases} \quad \text{mit} \quad t_0 = t_{0-} = 0. \quad (7)$$

Das kombinierte Signal setzt sich aus der Summe aller Konstanten und Sinus-Funktionen zusammen:

$$u(t) = \sum_{j=1}^2 \sum_{i=1}^s u_{j,i}(t), t \in \{0, \dots, (N-1)T_0\}. \quad (8)$$

Hierbei stellt s die Anzahl der stückweise konstanten Teilstücke eines APRBS mit der Länge N für eine Abtastzeit T_0 dar. Der Übertrag in ein zeitdiskretes Signal erfolgt durch $t = kT_0$.

In Bild 7 ist ein APRBS mit starker Beschränkung der Änderungsrate dargestellt. Durch die Nebenbedingung $|\dot{u}_{\max}| < \infty$ reduzieren sich die Sequenzen der stückweise konstanten Bereiche. Daher kann die maximale Änderungsrate nicht unabhängig von der minimalen Haltedauer T_H gewählt werden. Sobald sich die Zeiten t_{i+} und $t_{(i+1)-}$ überschneiden, ergibt sich eine mehrdeutige Funktion für das sinAPRBS. Anschaulich bedeutet dies, dass die konstanten Bereiche nicht erreicht werden können.

Durch (5) wird ersichtlich, dass die Frequenzen der Sinusübergänge von den jeweiligen Amplitudendifferenzen des ursprünglichen APRBS abhängt. Dies wird auch durch Bild 7 verdeutlicht. Um die maximale Veränderungsrate einzuhalten, ergibt sich eine niedrigere Frequenz für den ersten Sprung. Der letzte hingegen ist deutlich hochfrequenter.

Wird das in Bild 5 verwendete Signal angepasst um eine maximale Verfahrensgeschwindigkeit von $|\dot{u}_{\max}| = 0.27$ einzuhalten, so ergibt sich das sinAPRBS, welches in Bild 8 zu sehen ist. Qualitativ hat sich der Verlauf des Signals in a) nur geringfügig geändert. Jedoch sind durch die Einbringung der Nebenbedingung nun Bereiche in den Bildern b)-d) nicht mehr erreichbar (grau hinterlegt). Am deutlichsten ist dies, wenn man auf den Unterraum von $u(k-1)$, $y(k-1)$ in d) blickt. Durch die Randbedingung ergibt sich eine diagonale bandförmige Datenverteilung. Die Begrenzung der Änderungsrate ist auch in b) in der Ableitung des Signals zu erkennen.

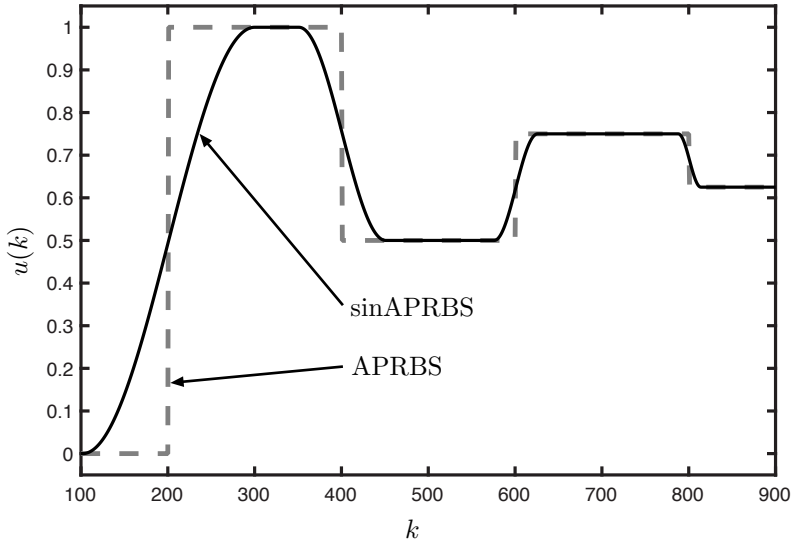


Bild 7: Anpassung eines APRBS zum sinAPRBS. Zur Darstellung wurde hier eine konstante Haltedauer $T_H = 200$ gewählt.

4 Ergebnisse

In diesem Abschnitt sollen die verschiedenen Signaltypen auf die erreichbare Approximationsqualität auf einem nichtlinearen Modell untersucht werden.

4.1 Trainingsdaten

Für jeden Signaltyp werden Datensätze mit dem in Bild 2 gezeigten Prozess erzeugt. Es wird hier angenommen, dass das System eine maximale Änderungsrate von $|\dot{u}_{\max}| = 0.27$ realisieren kann. Daher wird hier neben den zwei Sinus-basierten Signaltypen der neu vorgestellte Signaltyp anstelle des klassischen APRBS verwendet. Für den Vergleich der Leistungsfähigkeit wird je ein Datensatz mit und einer ohne Rauschen am Prozessausgang erzeugt. Im verrauschten Fall wird ein normal verteiltes, weißes Rauschsignal $n(k)$ mit Standardabweichung $\sigma_n = 0.02$ verwendet.

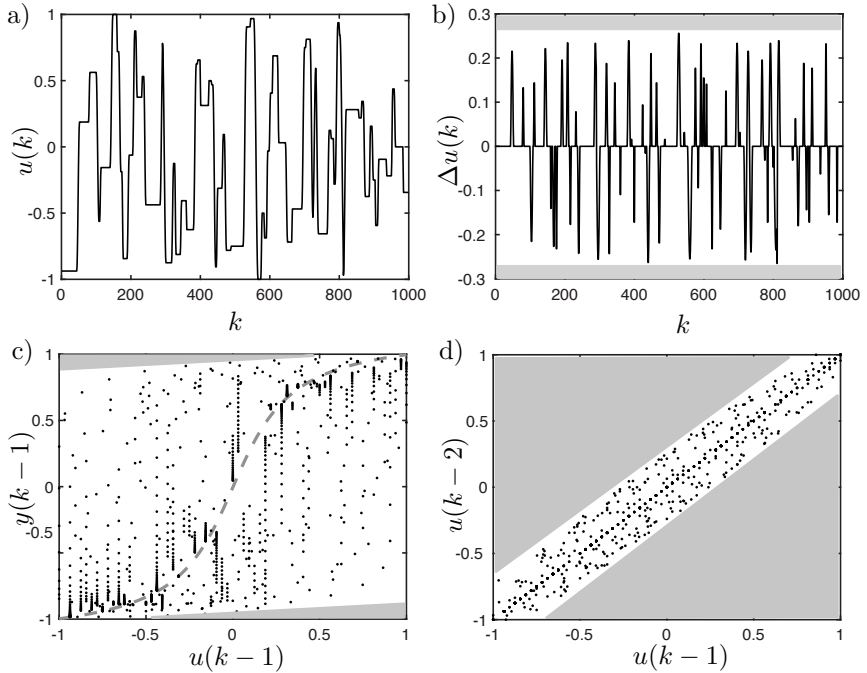


Bild 8: a) Signalverlauf eines sinAPRBS.

b) zeitliche Ableitung.

c) Eingangsraum für ein NARX Modell 1. Ord., Statik grau gestrichelt.

d) Eingangsraum bestehend aus verzögerten Prozess Eingängen.

4.2 Trainingsalgorithmus

Für die Identifikation von nichtlinearen dynamischen Prozessen haben sich lokale Modellnetze (LMN) in vielen Bereichen etabliert. Speziell die Aufteilung der Eingänge in Regel-Prämissen (z) und Regel-Konklusionen (x) erweist sich in vielen Bereichen als vorteilhaft [2]. Durch die Unterteilung des z -Eingangsraums können die lokalen Modelle als Linearisierung des Prozesses im Bereich der Gültigkeit des lokalen Modells verstanden werden.

Hier soll der hierarchische Baumkonstruktionsalgorithmus HILOMOT (vgl. [2]) zur Erzeugung von LMNs eingesetzt werden. Durch die inkrementelle Unterteilung des z -Eingangsraums wird eine schrittweise Anpassung

des Modells an den Prozess erreicht. Die Schnittrichtung wird hier auf Basis des Simulationsfehlers optimiert. Im Gegensatz zur Einschnittprädiktion in der sich der Eingangsraum aus Prozessein- und -ausgängen zusammensetzt ($\underline{z}(k) = [u(k-1) \ y(k-1)]$), werden bei der Simulation die Modellausgänge im z -Eingangsraum verwendet: $\underline{z}(k) = [u(k-1) \ \hat{y}(k-1)]$.

Um eine Überanpassung des Modells an die gemessenen Daten zu verhindern, wird die Anzahl der lokalen Modelle des gewählten Netzes entsprechend des minimalen korrigierten Akaike Information Criteria (AIC_C) gewählt.

4.3 Partitionierung des Eingangsraums

Die Unterteilung des z -Eingangsraums beeinflusst maßgeblich die Simulationseigenschaften des identifizierten Modells. Bereiche in denen sich Eigenschaften des Prozesses stark ändern, müssen deutlich feiner unterteilt sein als solche, welche näherungsweise linearen Einfluss haben. In Bild 9 ist die Unterteilung des Eingangsraums für die drei verschiedenen Signale mit Prozessrauschen ($\sigma_n = 0.02$) zu sehen.

Es hat sich bereits in Bild 5 und 4 für das APRBS und das Multisinus-Signal eine gleichmäßige Datenabdeckung gezeigt. Daher ergibt sich durch das zusätzliche Rauschen und die Begrenzung der Verfahrensgeschwindigkeit keine wesentliche Änderung und weiterhin bei beiden Signalen eine dichte Punktverteilung und damit hochauflösende Unterteilung des Eingangsraums. Das Chirp-Signal hingegen bietet nur eine geringe Datendichte im mittleren Bereich des von z aufgespannten Raums. Daraus folgt eine nur grobe Partitionierung in diesem Gebiet.

Die grobe Unterteilung bei der Verwendung eines Chirp-Signal spiegelt sich auch in der approximierten statischen Kennlinie des Prozesses wieder (Bild 9 unten rechts). Hier ergibt sich eine starke Abweichung der Verstärkung des Modells im Vergleich zum Prozess. Im Gegensatz hierzu können das Multisinus und das sinAPRBS die Kennlinie qualitativ gut beschreiben.

4.4 Simulationsfehler auf Testdaten

Beim Vergleich verschiedener Eingangssignale mit Hinblick auf die Simulationsqualität eines darauf identifizierten Modells, kommt der Wahl des Testsignals eine entscheidende Rolle zu. Wird der Testsignaltyp aus den

untersuchten Signalklassen für das Anregungssignal gewählt, so hat dieser Signaltyp tendenziell einen Vorteil gegenüber den anderen. Um diesem Konflikt zu entkommen, wird das Testsignal mit gleichen Anteilen aus allen untersuchten Signaltypen zusammengesetzt und im Anschluss auf dem Prozess simuliert. Somit besteht der Testdatensatz für jedes Modell zu je einem Drittel aus Trainingsdaten. Nach Ansicht der Autoren, ist dies eine faire Möglichkeit die Leistungsfähigkeit der Signale zu beurteilen.

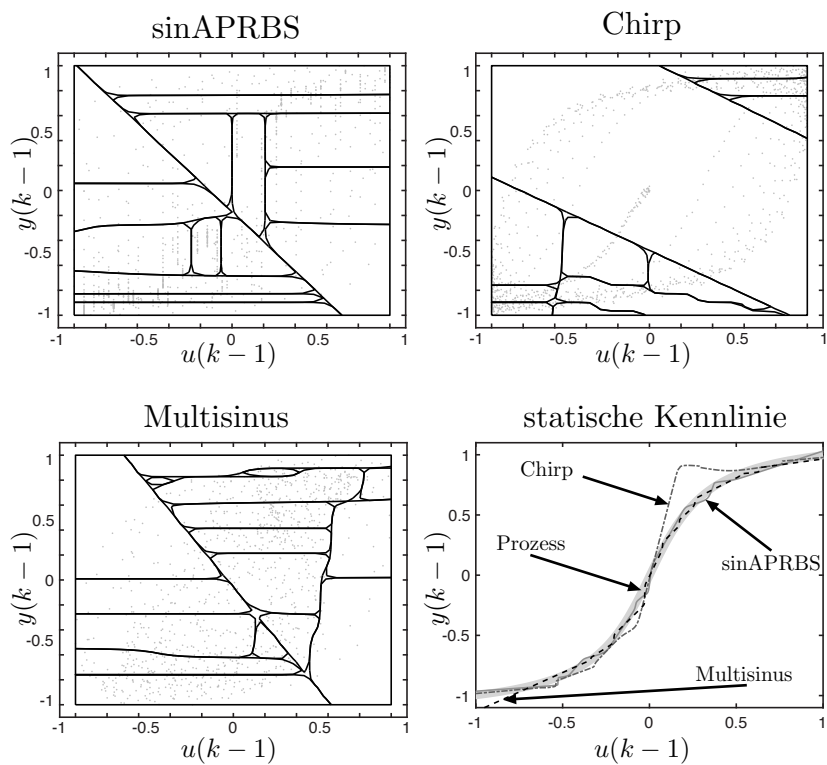


Bild 9: Partitionierung des z -Eingangsraums mit den gemessenen Daten (graue Punkte). Die approximierten und die echte statische Kennlinie des Prozesses ist unten rechts.

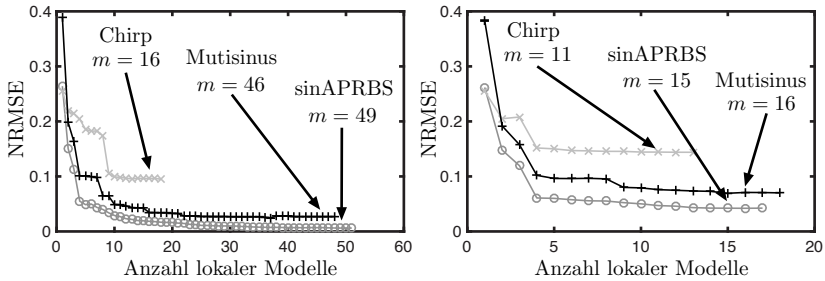


Bild 10: Fehler auf Testdaten für den unverrauschten Fall (links) und den verrauschten Fall (rechts).

In Bild 10 sind die Konvergenzverläufe der trainierten Modelle zu sehen. Die gewählte Modellkomplexität ist entsprechend gekennzeichnet. Da alle *Testdaten*verläufe deutlich in eine Sättigung gehen, erscheint das AIC_C als Abbruchkriterium als vernünftige Wahl. Als Fehlermaß für den Vergleich wird der *normalized root mean squared error* (NRMSE) auf Simulationsdaten gewählt. In Tabelle 1 sind die Trainings- und Testdatenfehler zu sehen. Hervorzuheben ist hier besonders der verrauschte Fall. Auf Trainingsdaten schneiden alle Signale vergleichbar ab. Auf Testdaten jedoch zeigen sich starke Unterschiede. Hervorzuheben ist, dass das sinAPRBS stets die beste Modellqualität liefert.

Tabelle 1: NRMSE der identifizierten Modelle auf Trainings- und Testdaten.

Datensatz	σ_n	Train	Test	σ_n	Train	Test
sinAPRBS	0	0.0035	0.0062	0.02	0.0512	0.0422
Chirp	0	0.0127	0.0963	0.02	0.0522	0.1443
Mutisinus	0	0.0033	0.0271	0.02	0.0448	0.0709

5 Zusammenfassung

Es wurden die gängigsten Signale für die Identifikation von nichtlinearen dynamischen Systemen kurz vorgestellt und wichtige Eigenschaften wie die Datenabdeckung im Eingangsraum herausgestellt. Es hat sich gezeigt, dass Sprung- und Sinus-basierte Signale gegensätzliche Vor- und Nachteile aufweisen. Dies motivierte die Vorstellung eines neuen Signaltyps sinAPRBS als Kombination von Sinus und APRBS. Anhand eines

simulierten nichtlinearen dynamischen Beispielprozesses wurde die Leistungsfähigkeit des neuen Signaltyps demonstriert. Es weist signifikante Vorteile auf hinsichtlich: (i) Eingangsraumabdeckung, (ii) Berücksichtigung von maximalen Änderungsraten (rate constraints), (iii) Genauigkeit des identifizierten Modells. Die Vorteile gelten sowohl für den unverrauschten als auch verrauschten Fall.

Literatur

- [1] W Baumann, S Schaum, K Roepke, and M Knaak. Excitation signals for nonlinear dynamic modeling of combustion engines. In *Proceedings of the 17th World Congress, The International Federation of Automatic Control, Seoul, Korea, 2008*.
- [2] Julian Belz and Oliver Nelles. Input selection using local model network trees. *IFAC Proceedings Volumes*, 47(3):4128–4133, 2014.
- [3] Michael Deflorian and Susanne Zaglauer. Design of experiments for nonlinear dynamic system identification. In *IFAC 18th World Congress, Milano 2011, 2011*.
- [4] Christoph Hametner, Markus Stadlbauer, Maxime Deregnaucourt, Stefan Jakubek, and Thomas Winsel. Optimal experiment design based on local model networks and multilayer perceptron networks. *Engineering Applications of Artificial Intelligence*, 26(1):251–261, 2013.
- [5] Lennart Ljung. *System identification*. Springer, 1998.
- [6] Oliver Nelles. *Nonlinear system identification*. Springer, 2001.
- [7] Oliver Nelles and Rolf Isermann. Identification of nonlinear dynamic systems classical methods versus radial basis function networks. In *American Control Conference, Proceedings of the 1995*, volume 5, pages 3786–3790. IEEE, 1995.
- [8] Johan Paduart, Lieve Lauwers, Jan Swevers, Kris Smolders, Johan Schoukens, and Rik Pintelon. Identification of nonlinear systems using polynomial nonlinear state space models. *Automatica*, 46(4):647–656, 2010.
- [9] Rik Pintelon and Johan Schoukens. *System identification: a frequency domain approach*. John Wiley & Sons, 2012.

- [10] Aleksander Preglej, Jakob Rehrl, Daniel Schwingshackl, Igor Steiner, Martin Horn, and Igor Škrjanc. Energy-efficient fuzzy model-based multivariable predictive control of a hvac system. *Energy and Buildings*, 82:520–533, 2014.
- [11] Daniel E Rivera, Hyunjin Lee, Hans D Mittelmann, and Martin W Braun. Constrained multisine input signals for plant-friendly identification of chemical process systems. *Journal of Process Control*, 19(4):623–635, 2009.
- [12] Johan Schoukens and Rik Pintelon. *Identification of linear systems: a practical guideline to accurate modeling*. Elsevier, 2014.
- [13] Johan Schoukens, Mark Vaes, and Rik Pintelon. Linear system identification in a nonlinear setting: Nonparametric analysis of the nonlinear distortions and their impact on the best linear approximation. *IEEE Control Systems*, 36(3):38–69, 2016.
- [14] Manfred Schroeder. Synthesis of low-peak-factor signals and binary sequences with low autocorrelation (corresp.). *IEEE Transactions on Information Theory*, 16(1):85–89, 1970.
- [15] Daniel Schwingshackl, Jakob Rehrl, and Martin Horn. Lolimot based mpc for air handling units in hvac systems. *Building and Environment*, 96:250–259, 2016.
- [16] Nils Tietze. Model-based calibration of engine control units using gaussian process regression. 2015.
- [17] Herbert JAF Tulleken. Generalized binary noise test-signal concept for improved identification-experiment design. *Automatica*, 26(1):37–49, 1990.
- [18] Patricio E Valenzuela, Cristian R Rojas, and Hakan Hjalmarsson. Optimal input design for non-linear dynamic systems: a graph theory approach. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 5740–5745. IEEE, 2013.

From Real World Data to Test Functions

Andreas Fischbach, Martin Zaefferer, Jörg Stork, Martina Frieze,
Thomas Bartz-Beielstein

SPOTSeven Lab, Dept. of Comp. Sci. and Eng. Sci.
TH Köln

E-Mail: {andreas.fischbach, martin.zaefferer, joerg.stork, martina.frieze,
thomas.bartz-beielstein}@th-koeln.de
www.spotseven.de

1 Introduction

When researchers and practitioners in the field of computational intelligence are confronted with real-world problems, the question arises which method is the best to apply. Nowadays, there are several, well established test suites and well known artificial benchmark functions available. However, relevance and applicability of these methods to real-world problems remains an open question in many situations. Furthermore, the generalizability of these methods cannot be taken for granted. Some preliminary ideas about generalizability are discussed in [1, 2].

This paper describes a data-driven approach for the generation of test instances, based on real-world data, as depicted in Figure 1. The test instance generation uses data-preprocessing, feature extraction, modeling, and parameterization. It was applied to several real-world scenarios, e.g., in the context of genetic programming [3]. In this work we apply this concept on a classical design of experiment real-world project and generate test instances for benchmarking, i.e., design of experiment methods and model fitness. But it can also be used to compare and analyze several surrogate techniques and optimization algorithms as well.

In most cases, complex and expensive real-world problems do not provide sufficient data for comparison of methods. Thus, our goal is our goal is to create a toolbox containing multiple data sets of real-world projects. With that toolbox, researchers are granted access to both the data sets and the derived test functions.

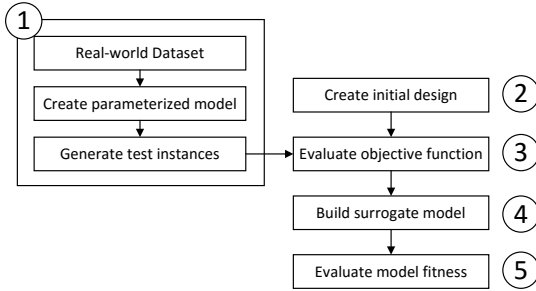


Figure 1: *Simplified test process of surrogate models fitted upon evaluation on initial designs with real-world data based test instances. An optional validation step can be added before the test instances are used in step 3.*

This work mainly focuses on the following questions:

- (Q-1) What is the characteristic of a variation of a certain model parameter in terms of the models fitness landscape?
- (Q-2) How can similarity between models be computed and what are useful thresholds to separate similar models from almost equal models on the one hand, and completely different models on the other hand?

Considering an example application for the test function generator, the design of experiment used to gather the reference data set used in this work will be further analyzed:

- (Q-3) Which design of experiment works best for the underlying real-world problem?

The remainder of this work is organized as follows. Section 2 gives a literature review. Section 3 describes the process of the test function generation and applied modeling techniques as well as model similarity measures used. Section 4 illustrates the reference data set and the example application, the evaluation of design of experiment methods. Section 5 concludes the work and gives a short outlook for future work.

2 Related Work

In the research field of benchmarking real-parameter problems many contributions address the random generation of problem instances based on user defined parameters, like the *Max-Set of Gaussian Landscape Generator* [4], or the *Krigifier* [5], both using Gaussian Processes.

The *Max-Set of Gaussian Landscape Generator* computes the upper envelope of m weighted Gaussian process realizations and can be used to generate continuous, bound-constrained optimization problems. The landscape generator is parameterized to control, e.g., the number of Gaussian components and implicitly, the number of local optima, the occurrences and variations of hills and the peaks, and the global optimum.

The *Krigifier* realizes a procedure for generating nonlinear objective functions. Its idea is based on the convenient supposition that objective functions are realizations of stochastic processes. The user specifies an underlying trend, a stochastic process and a finite number of points at which the process will be observed. The Krigifier creates a noise term and uses the trend and the noise term to produce an objective function.

A completely different approach is addressed by the *Real-Parameter Black-Box Optimization Benchmarking* [6]. The organizers of this benchmarking challenge choose and implement a benchmarking function testbed, typically covering artificial test functions like Sphere, Rosenbrock, Rastrigin, etc. Participants then have to apply their black-box problem solvers and their results will be gathered and compared.

All these approaches do not rely directly on real-world problems. Our work realizes one step towards closing a gap in means of validating methods regarding their applicability and generalizability in practical deployment.

3 Test Function Generation

The simplified process of real-world data based test function generation is depicted in Figure 2. It consists of the following steps:

1. Process a data set of a real-world problem, \mathbf{X} is the design matrix and \mathbf{Y} is a vector of corresponding outcome values of the underlying process.

2. Build a model, further denoted as *level 0 model*, that is suitable for regression and interpolation purposes of the data. In this work the Kriging technique is used to build the model.
3. Create a parameter α to vary the previously fitted *level 0 model*. The parameter α is a scalar or vector, that perturbs the generated model. It may, e.g., define a change in parameters or other variables of the derived model. First, bounds are generated for α , ensuring numerical robustness. Then, n instances α_i with $i = 1, \dots, n$ are randomly created within the chosen bounds. Finally, a randomly selected subset of the desired number of $m < n$ instances is chosen. This ensures that the instances which need to be solved by the evaluated methods, are never known in advance.
4. Apply the selected α instances on the *level 0 model* to retrieve the desired number of m *level 1 model* instances, each coupled with an α_j , $j = 1, \dots, m$. The process ensures the fulfillment of the requirements on the similarity demands of the models by computing similarity measures and discarding infeasible α instances.

The remainder of this section describes the Kriging modeling technique, the model variation and the model similarity evaluation.

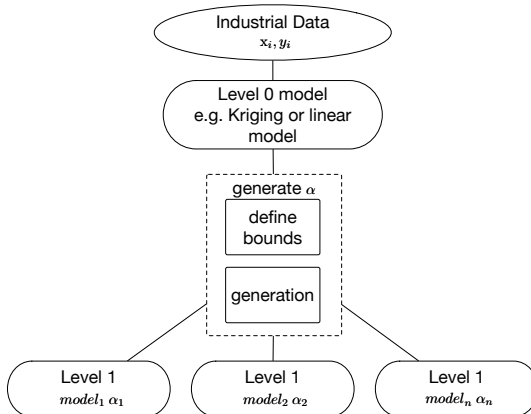


Figure 2: *Resulting hierarchy of the generation process of different models to be used as test function instances.*

3.1 Kriging

Often used for the purpose of regression and interpolation, Kriging is a modeling method based on Gaussian processes. In the following we will stick closely to the descriptions by Forrester et al. [7]. Further details can be found in their book. Given a set of n solutions $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1\dots n}$ in a k -dimensional continuous search space with observations $\mathbf{y} = \{y^{(i)}\}_{i=1\dots n}$, Kriging tries to determine an expression for a predicted value at an unknown location by interpreting the observations \mathbf{y} as realizations of a stochastic process. The stochastic process is defined by the set of random vectors $\mathbf{Y} = \{Y(\mathbf{x}^{(i)})\}_{i=1\dots n}$. The correlation of the random variables $Y(\cdot)$ is modeled as follows [7]:

$$\text{cor} \left[Y(\mathbf{x}^{(i)}), Y(\mathbf{x}^{(l)}) \right] = \exp \left(- \sum_{j=1}^k \theta_j |x_j^{(i)} - x_j^{(l)}|^{p_j} \right). \quad (1)$$

The matrix that collects correlations of all pairs $\{(i, l)\}$ is called the correlation matrix Ψ . It is used in the Kriging predictor

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \boldsymbol{\psi}^T \Psi^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}), \quad (2)$$

where $\hat{y}(\mathbf{x})$ is the predicted function value of a new sample \mathbf{x} , $\hat{\mu}$ is the maximum likelihood estimate (MLE) of the mean and $\boldsymbol{\psi}$ is the vector of correlations between training samples \mathbf{X} and the new sample \mathbf{x} . The width parameter $\boldsymbol{\theta} = (\theta_1, \dots, \theta_j, \dots, \theta_k)^T$ determines how far the influence of each sample point \mathbf{x} spreads. In detail, the larger the width parameter is, the faster are the potential changes in the predicted value. The smaller the width parameter is, the slower are the potential changes in the prediction. Since there is one θ_i for each dimension, this parameter can control the activity in each dimension. The parameter p_j is usually fixed at $p_j = 2$, and defines the shape of the correlation function: At $p_j = 2$, the correlation function is more smooth, whereas $p_j = 1$ is less smooth. In case of noise, the parameter λ is added to the diagonal of the correlation matrix Ψ . This allows the model a more smooth fit through observations (regression), in contrast to the default which reproduces all training data exactly (interpolation). Classically, λ is used to deal with noisy data. But it can as well be used to smoothen more rugged fitness landscapes.

All model parameters are determined by Maximum Likelihood Estimation (MLE). For $\boldsymbol{\theta}$, λ and \mathbf{p} , MLE requires numerical optimization.

3.2 Model variations

The main parameters controlling the behavior of the model are θ and λ . The major goal of the test function generator is the deployment upon real-world data, which is usually noisy. So the variation of the parameter λ is a natural choice at a first glance. In addition the variation of the width parameter θ seems important to change the model (slightly) by maintaining the general characteristic of the fitness landscape under certain circumstances. The bounds of the variations of the parameters have to be defined carefully, otherwise the model can show signs of degeneration.

The test function generator will compute lower and upper bounds for α according to the fitted *level 0 model*. The first component of α represents the λ value and the remaining components represent the corresponding θ values. They will be added to their corresponding values of the *level 0 model* to retrieve the altered *level 1 model*. Afterwards the correlation Matrix Ψ is recalculated, so that the changes take effect before predictions are made. The test function generation returns the desired number of similar functions, randomly drawn from the search space defined by the *level 0 model* and the bounds for α . If the given bounds do not allow the creation of sufficient feasible instances, an error message is produced.

An example is shown in Fig. 3. Here, the one-dimensional function

$$f(x) = (-18x - 2)^2 \sin(20x - 4)$$

is first sampled by 11 equidistant points. The derived level 0 Kriging model has $\lambda = 0$ and $\theta = 100$. To derive the level 1 models, the bounds for *alpha* are set to $alpha_{low} = [0, -90]$ and $alpha_{high} = [1, 900]$ so that λ will be set to values between zero and one, and θ between ten and one thousand. For demonstration purposes, the extreme values for *alpha* are chosen, as shown above each plot in Fig. 3. The plot shows that different α values affect the ruggedness of the function and vary the number of local optima.

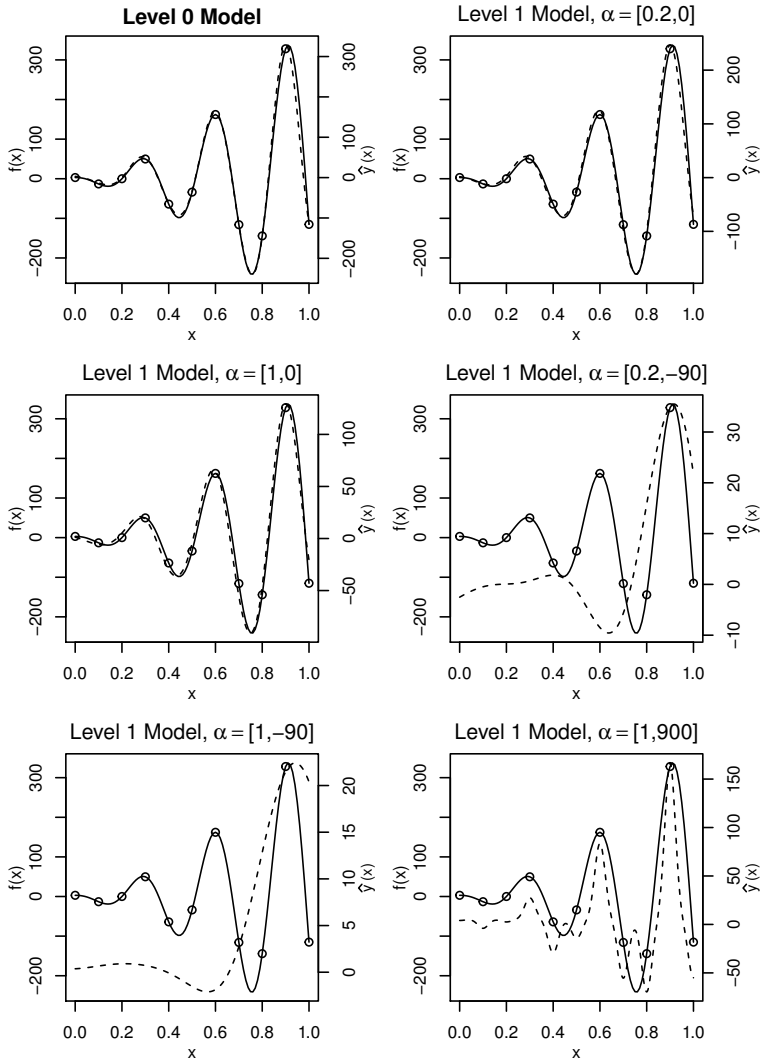


Figure 3: A level 0 Kriging model and several derived level 1 models. The solid line is the true function $f(x)$, circles indicate observations used to fit the level 0 model and the dashed line indicates the level 1 model $\hat{y}(x)$.

3.3 Model Similarity Computation

In addition to the quality of a model itself a measurement of the similarity of two models is needed. This should prevent the generation of test functions which are not anymore correlated to the given training data. Some correlation is desirable, so that the generated instances reflect the structure of the real-world data to some degree.

The *level 1 models* difference from the *level 0 model* will be computed evaluating the models at a pre defined equidistant grid and taking the differences of the predicted values of the models. It must be ensured that there is at least a difference large enough to ensure that the models are not equal or almost equal, otherwise we are always looking at the very same function. On the other hand, the difference must not be too large, because the characteristic of the underlying problem must be preserved, otherwise the results of the evaluations are questionable.

Each of the accuracy measures discussed in the following is unique and none can be considered as superior to the others. One goal in this work is to define an ensemble of measurements with carefully defined thresholds. This should lead to a confident set of different models comparable to the underlying real-world problem.

- The **mean absolute error** (MAE) is a popular measure to evaluate model fitness. It is defined as follows:

$$MAE = \frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{n}$$

- The **root mean square error** (RMSE) has been used as a standard metric to measure model performance in different scientific fields like meteorology, air quality, and climate research studies [8]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n |\hat{y}_i - y_i|^2}{n}}$$

- **Pearsons correlation coefficient** (r) is a measure of the amount of linearity between two continuous variables X and Y , giving a value between -1 and $+1$ inclusive, where $+1$ is total positive correlation, 0 is no correlation, and -1 is total negative correlation. Regarding the prediction values of the *level 0 model* and *level 1 model*, correlation

values are desired to be closer to +1 as to 0. If they are too close to +1, they can be discarded due to their degree of conformity, while correlation values near 0 or even negative correlations indicates that the two models are not similar anymore.

- To be capable of dealing with non-linear models or non-linear changes of the models, rank based correlation is regarded as well. In this work **Spearman's** ρ rank correlation coefficient is used. An increasing coefficient value describes an agreement of the rankings. A value of +1 implies that the rankings are the same, -1 that the rankings are inverse and 0 that the rankings are independent.
- In addition a **t-test** will be performed to compare the two populations of predictions. The p-value of the t-test describes the likeliness of the observed data assuming the null hypothesis is true. The null hypothesis is that the compared data groups are from the same population. For two similar models a high p-value next to +1 is to be expected. It has to be regarded that t-test demands several assumptions, i.e., the normal distribution of the data.

Although these statistics have been used for years now, there is no consensus which of these statistics is superior. Measures of absolute values like the MAE or the RMSE lack interpretability, while statistics delivering values between -1 and +1 or 0 and +1 are easy to interpret if the distribution of the observed values is appropriate. As a conclusion, a combination of these statistics will be used.

The interface of the test function generator provides a function that takes training data, validation data, lower and upper bounds for the variation vector α and the number of test functions to generate. These instances are drawn randomly in the search space defined by the *level 0 model* and α . The test function generator performs the similarity computation automatically and discards generated models that are within the bounds for α but can not be seen as similar or are too similar according to the previously discussed statistics.

4 Example Application - DoE Evaluation

The test function generator was motivated by two different kinds of applications. The first application is to analyze the generalizability of a method developed for a specific problem. The second application is to further analyze a specific real world data set. The remainder of this section introduce the real-world data set, the experimental setup and the results of the experiments.

4.1 Reference Data Set

The industrial data, which depicts the basis for our test function generator, originated from experiments to optimize the development of gas sensors for combustion processes [9]. In a combustion process, a number of different gases are produced. A fraction out of these gases are crucial indicators for the goodness of the combustion and important for the process control. Therefore, the target of the new sensor design was to improve the sensor sensitivity for a specific gas, while reducing the cross sensitivity to other gases. The experiments were conducted using a mixture of seven different gases as input, further denoted as X_1 to X_7 . Moreover a measurement in voltage of two different sensors were retrieved as output, further denoted as Y_1 and Y_2 . The underlying experimental design was a response surface design fitted to built the associated regression model, including two-way interactions and quadratic behavior.

The design size was set to 80 experiments as a result of time-consuming real-world measurements. So, all experiments could be performed within one week. An overview of the data is outlined in Table 1. The data was anonymized and standardized due to confidentiality restrictions.

An additional 60 experiments were run, with the same characteristics as described above. The resulting dataset was used for model validation.

These two data sets were previously used to compare and analyze advantages and disadvantages of several different regression modeling techniques on sparse and limited data, e.g. Bayesian robust linear regression, standard regression approaches, like ordinary least squares and **Lasso** (least absolute shrinkage and selection operator, see [10]), and genetic programming [9]. The results show that the **Lasso** regression model performs best on

Table 1: Overview of the standardized dataset used to generate the models of the sensors. Each input of the model is denoted by an X and each sensor output is denoted by an Y .

	X1	X2	X3	X4	X5	X6	X7	Y1	Y2
Minimum	-1.13	-1.21	-1.16	-1.13	-1.15	-1.17	-1.00	-1.94	-2.06
1st Quartal	-1.13	-1.21	-1.16	-1.13	-1.15	-1.17	-0.82	-0.63	-0.58
Median	0.09	0.03	0.12	0.08	0.08	0.05	-0.39	0.06	0.09
Mean	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3rd Quartal	1.30	1.26	1.40	1.29	1.28	1.28	0.59	0.66	0.67
Maximum	1.30	1.26	1.40	1.29	1.31	1.28	3.79	2.32	2.28

the validation data set predicting $Y1$. In the resulting model only the parameters $X1$ to $X4$ were included.

Due to the production process of the sensors, future experiments will most likely lead to similar linear regression models. The customer is heavily interested in these easy to understand models and from the research perspective it is very important and interesting to analyze the capabilities of the applied design of experiment. Therefore the test function generator will be set up to deliver similar problem instances based on the given data.

4.2 Experimental setup

As a first step to demonstrate the application of the test function generator, the variation of the Kriging parameters λ and θ will be analyzed. We use the data described in the previous section limited to the inputs $X1$ to $X4$, which are the most important and interesting parameters according to the results of the different modeling techniques [9].

The bounds for α for this screening experiment are given in Table 2. All experiments are performed using the free software environment for statistical computation, R⁵. As optimizer for the estimation of the Kriging model parameters θ and λ , Global Optimization by Differential Evolution (DEoptim), with a budget of 1000 evaluations is set.

To analyze the generalizability of the design of experiment created to obtain the training data, further referred to as *base design* we apply the

⁵R as well as all employed packages are available at <http://cran.r-project.org/>

Table 2: Lower and upper bounds for λ and θ and the nr. of values (steps) taken in each dimension to build a grid as design of experiment to evaluate the influence of λ and θ variations on the model similarity.

	lambda	theta
lower bound	0.1	-2
upper bound	1	10
steps	4	4

test function generator on the data set. The generator will be set up to generate modified Kriging models of the obtained *level 0 model* with previously adjusted similarity thresholds.

Several different designs will be employed and linear regression *level 2 models* fitted by means of the least squares method by evaluating the design points on the drawn *level 1 model* instances. The model fitness will be evaluated by computing the RMSE between the *level 1 models* and *level 2 models* at a predefined grid. These experiments are based on previous work to compare different design methods, e.g., random sampling, Latin hypercube sampling and full factorial designs, see also [11]. The test function generator is set up to generate 20 different *level 1 models*.

Several designs will be set up to evaluate the models. The *base design* will be pre processed, as the dispensable columns for the parameter X_5 to X_7 will be omitted and remaining duplicates will be removed. This results into a design size of 67 points. It has to be regarded that an optimized design created for the parameters X_1 to X_4 can look different.

This design will be compared with a Full factorial design (FFD) with 2 levels per factor (16 points) and a FFD with 3 levels per factor (81 points) as well. In addition Latin hypercube designs (Lhd) with 16, 67 and 81 points resp. and simple uniform random sampling designs of corresponding sizes will be created.

Each design will be used to retrieve predictions of the 20 different *level 1 models* and fit a linear regression model of the following form, which will be taken as a benchmark model structure:

$$\hat{y} = \beta_0 + \sum_{i=1}^4 \beta_i x_i + \beta_{14} x_1 x_4 + \beta_{34} x_3 x_4.$$

Table 3: Summary of a linear model fit, showing the estimated influence and significance of the different parameters on the RMSE.

	Estimate	Std. Error	t value	p value	
(Intercept)	0.2029	0.0032	62.88	< 2e-16	***
lambda	0.1099	0.0047	23.32	<2e-16	***
theta1	-0.0013	0.0007	-1.83	0.0669	.
theta2	-0.0042	0.0007	-5.96	3.55e-09	***
theta3	-0.0140	0.0007	-19.95	<2e-16	***
theta4	-0.0093	0.0007	-13.33	<2e-16	***

The resulting models will be evaluated at a pre defined equidistant grid and the RMSE between the predictions of the *level 1 models* and the *level 2 models* will be computed.

4.3 Results

At first the results of the screening experiments of the variations of α will be discussed. In total, 1045 experiments with different α instances were run, resulting from the 4 different values per parameter ($4^5 = 1024$) plus additional 20 runs of one factor at a time variations and one trial with each component in α set to 0 for validation purpose.

Results of the one factor at a time variations are shown exemplarily in Figures 4 and 5 for λ , θ_1 and θ_4 . It can be seen that, even with increasing RMSE and MAE, the correlation coefficients seem insensitive against changes of one parameter, while the p value of the t-test seems very sensitive against changes. Another interesting aspect is, that the p value seems less sensitive for changes of λ than for the changes of the θ values.

In Figure 5, it can be seen, that the change of the variation of θ_4 from 2 to 4 only results in a small change of the RMSE and MAE. This sometimes can occur when changes of α are cut to the upper or lower bounds.

As expected, a linear model fit revealed that almost all components of α have a significant effect on the RMSE, see Table 3. Similar results were obtained for the other measures, i.e., MAE, Pearsons correlation coefficient r , Spearmans correlation and the p value of the t-test. The p value of the t-test seems to be the most sensitive of the measures.

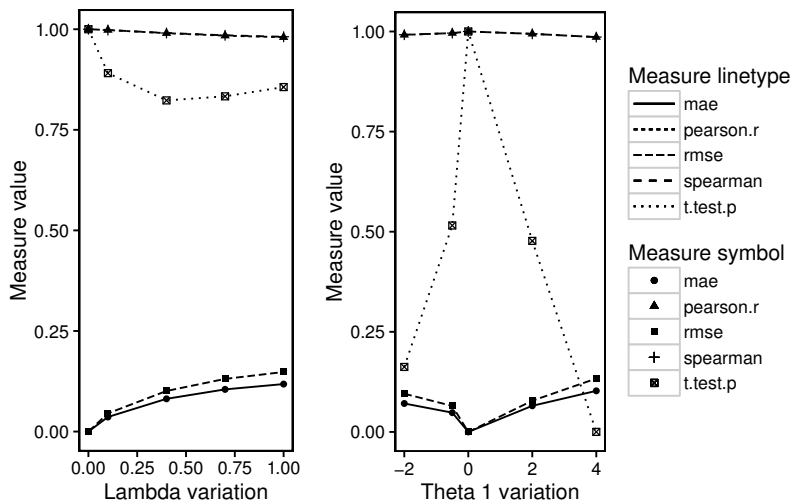


Figure 4: *Effects of the variation of model parameters on the chosen similarity measures MAE, RMSE, Pearson's r, Spearman and the t-test p value. **Left:** Effects of λ changes. **Right:** Effects of θ_1 changes.*

A detailed look at the result table from the screening experiment was taken, to decide which thresholds for each of the measures should be taken to discard test function instances. In the summary of the measures, depicted in Table 4, the distribution of the measures can be seen. The correlation coefficients does not become very low, with their mean at around 0.95. The p value of the t-tests distributes on the whole scale from 0 to +1.

A full linear model with interaction terms between all parameters and main effects up to second degree predicting the RMSE and Spearman correlation coefficient reveals an adjusted coefficient of determination of about 0.85 and 0.74 respectively.

For the design of experiment application, the bounds for the measures were set as shown in Table 5. With these thresholds 20 random test functions were drawn. The impact of the chosen designs on the RMSE of the derived linear models compared to the test function are shown in Figure 6. It can be seen that the base design, applied to create the training data in the real-world project, could not be outperformed by any other design type. At a first glance it might be surprising, that designs with a larger design size, e.g. the *Uniform 81* or even the *LHD 81*, perform worse. But these designs

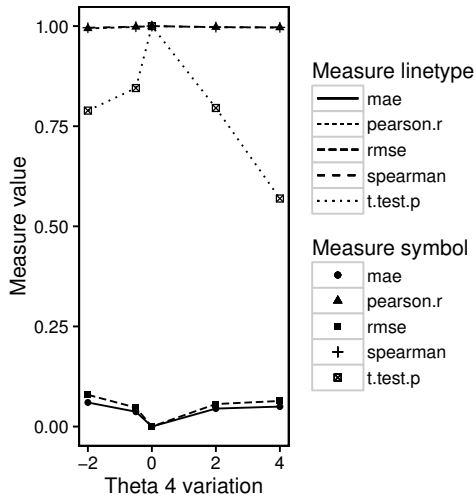


Figure 5: *Effects of the variation of θ_4 .*

were not set up to fit second order linear regression models and would therefore not be the first choice in such a setup. The results would surely look different if the *level 2 model* would as well be a Kriging model.

5 Conclusion

In this work the generation process of test function instances based on a real-world industrial data set is described. Based on this data a Kriging model is fitted and altered according to a variation parameter α , by adding its components to the corresponding model parameter. The generated instances will be discarded or kept according to thresholds for several model similarity measures. This leads finally to a test function instance pool that can be used to benchmark, i.e, design of experiment and modeling methods, for their practical use on the underlying problem.

Table 4: *Summary of statistical measures of similarity gathered on screening experiments for α .*

rmse	pearson.r	spearman	t.test.p
Min. :0.0000	Min. :0.8732	Min. :0.8560	Min. :0.0000006
1st Qu.:0.1819	1st Qu.:0.9268	1st Qu.:0.9283	1st Qu.:0.0189555
Median :0.2430	Median :0.9524	Median :0.9564	Median :0.1431840
Mean :0.2376	Mean :0.9486	Mean :0.9506	Mean :0.3314180
3rd Qu.:0.2935	3rd Qu.:0.9729	3rd Qu.:0.9749	3rd Qu.:0.6384715
Max. :0.4120	Max. :1.0000	Max. :1.0000	Max. :1.0000000

Table 5: *Thresholds for measures preventing too similar and degenerated test functions.*

	rmse	pearson.r	spearman	t.test.p
lower bound	0.2	0.85	0.85	0.5
upper bound	0.35	0.92	0.92	0.9

The major research Questions addressed in this work were:

(Q-1) What is the characteristic of a variation of a certain model parameter in terms of the models fitness landscape?

The variation of the Kriging model parameters λ and θ lead to altered models that were still correlated at rather high coefficient rates around 0.9. This was ensured by the definition of lower and upper bounds relative to the parameter values of the base *level 0 model*. This kind of limit the amount of changes applicable to the model and can be problematic for example when a large number of test instances is needed.

(Q-2) How can similarity between models be computed and what are useful thresholds to separate similar models from almost equal models on the one hand, and completely different models on the other hand?

This can kind of simple be visually analyzed for one or two dimensional problems, but is getting harder in general for larger number of dimensions. Statistical measures can help to distinguish models that are too similar or too different. Further analysis of the resulting fitness landscapes in sense of, e.g., number of local optima or gradients, can help to judge the similarity of two different models.

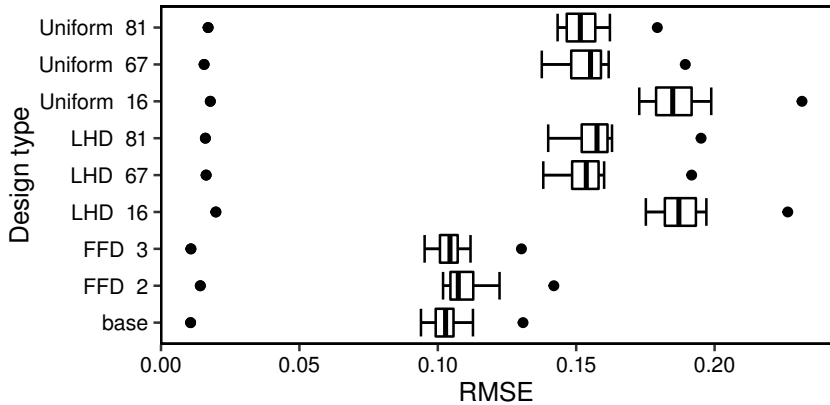


Figure 6: Resulting RMSE values of models based on different design of experiment methods each applied on the same set of 20 random test functions.

(Q-3) Which design of experiment works best for the underlying real-world problem

The application of different design of experiment methods has shown, that the base design, used to gather the training data set, was not outperformed by any other design method. Even larger designs like a Full factorial designs at three levels per factor and Latin hypercube designs with 81 points could not dominate the results. This of course has to be further analyzed by altering the model techniques for the *level 2 models*. Even slight changes to the base design can now be analyzed and might lead to interesting new design points for the customer in future real-world experiments.

Interesting future work include the shift of the similarity computation to the beginning of the instance generation. The Matrix computations to alter Kriging models can be regarded as time consuming, especially with high dimensional data. So it would be beneficial to compute an expectational value for the similarity of two models.

In addition, fitness landscape analysis is an interesting topic to include in the work. The fitness landscapes comparison based on the RMSE or correlations of function values can be extended by interesting features to add or to avoid, e.g., the number of local optima, plane areas or large gradients.

Finally, different variation methods on the models, e.g. rotating, scaling, distortion of the input space should be applied. Considering Kriging models, conditional simulation, could be applied to deliver possible realizations of a Gaussian process with a certain probability and therefore be very suitable to deliver model variations based on real-world data.

References

- [1] M. Chiarandini and Y. Goegebeur, “Mixed Models for the Analysis of Optimization Algorithms,” in *Experimental Methods for the Analysis of Optimization Algorithms* (T. Bartz-Beielstein, M. Chiarandini, L. Paquete, and M. Preuss, eds.), pp. 225–264, Germany: Springer, 2010.
- [2] T. Bartz-Beielstein, “How to Create Generalizable Results,” in *Springer Handbook of Computational Intelligence*, pp. 1127–1142, Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.
- [3] O. Flasch, “A Modular Genetic Programming System,” May 2015. Dissertation zur Erlangung des Grades eines Doktors der Ingenieurwissenschaften der Technischen Universität Dortmund, Fakultät für Informatik.
- [4] M. Gallagher and B. Yuan, “A general-purpose tunable landscape generator,” *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 590–603, Oct. 2006.
- [5] M. W. Trosset, I. for Computer Applications in Science, and Engineering., *The Krigifier [microform] : a procedure for generating pseudorandom nonlinear objective functions for computational experimentation / Michael W. Trosset*. Institute for Computer Applications in Science and Engineering, NASA Langley Research Center ; National Technical Information Service, distributor Hampton, VA : Springfield, VA, 1999.
- [6] N. Hansen, A. Auger, S. Finck, and R. Ros, “Real-parameter black-box optimization benchmarking 2009: Experimental setup,” Tech. Rep. RR-6828, INRIA, 2009.
- [7] A. Forrester, A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling*. Wiley, 2008.

- [8] T. Chai and R. R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature,” *Geoscientific Model Development*, vol. 7, pp. 1247–1250, June 2014.
- [9] M. A. Rebolledo Coy, S. Krey, T. Bartz-Beielstein, O. Flasch, A. Fischbach, and J. Stork, “Modeling and Optimization of a Robust Gas Sensor,” in *Bioinspired Optimization Methods and their Applications* (G. Papa and M. Mernik, eds.), pp. 267–278, May 2016.
- [10] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.
- [11] A. Fischbach, J. Stork, M. Zaefferer, S. Krey, and T. Bartz-Beielstein, “Analyzing Capabilities of Latin Hypercube Designs Compared to Classical Experimental Design Methods ,” in *25. Workshop Computational Intelligence* (F. Hoffmann and E. Hüllermeier, eds.), pp. 255–270, 2015.

Neuronale Netze vs. Support-Vektor-Maschinen – Ein direkter Vergleich der Klassifikationsmethoden für komplexe biologische Daten

Maria Trommer¹, Andreas Wenzel^{1,2}, Christian Walther^{1,2}

¹Hochschule Schmalkalden, Fakultät Elektrotechnik,
Forschungsgruppe Eingebettete Diagnosesysteme
Blechhammer 9, 98574 Schmalkalden

E-Mail: m.trommer@hs-sm.de, a.wenzel@hs-sm.de, c.walther@hs-sm.de

²Fraunhofer Institutsteil Angewandte Systemtechnik des IOSB
Am Vogelherd 50, 98693 Ilmenau

Zusammenfassung

Dieser Beitrag zeigt einen direkten Vergleich von zwei Klassifikationsmethoden an einem konkreten Beispiel. Es wird die Leistungsfähigkeit von Populationen von künstlichen Neuronalen Netzen (KNN) mit der Leistungsfähigkeit der Support-Vektor-Maschine (SVM) am Beispiel einer schlafstadienspezifischen Bewertung von EEG-Daten verglichen. Diese Daten stellen komplexe biologische Daten dar. Beide Methoden wurden eingehend auf die vorliegenden Daten optimiert. Dabei zeigten sich die grundlegenden Unterschiede zwischen den Methoden. Während sich die Komplexität der KNN einstellen lässt, ist dies bei der SVM nur eingeschränkt möglich. Die These, die SVM sei den künstlichen Neuronalen Netzen überlegen, konnte in diesem Beitrag für komplexe biologische Daten nicht bestätigt werden. Die Resultate zeigen zum einen eine höhere Instabilität der SVM-Klassifikatoren durch eine höhere Streuung der Ergebnisse. Zum anderen wird deutlich, dass die KNN in den meisten Gütekriterien bessere Werte erreichen können.

1 Einführung

Die Support-Vektor-Maschine (SVM) ist eine anerkannte Klassifikationsmethode und wurde bereits in vielen wissenschaftlichen Arbeiten für die verschiedensten Anwendungsgebiete erprobt. Die SVM wird hierbei als leistungsfähige Methode beschrieben. Schölkopf schreibt dazu in [1]: “At the same time, successful applications have demonstrated that SVMs not only have a more solid foundation than artificial neural networks, but are able to serve as a replacement for neural networks that perform as well or better, in a wide variety of fields.” Auf Grundlage dessen wird die These aufgestellt:

„Die SVM ist den künstlichen Neuronalen Netzen in ihrer Klassifikationsleistung überlegen.“

Künstliche Neuronale Netze (KNN) sind eine der bekanntesten Klassifikationsmethoden und intensiv erforscht, speziell auch für die Anwendung auf biologische Daten. So wurden in [2] die KNN für die Klassifikation von Schlaf anhand von EEG-Daten optimiert und eingesetzt. Die Resultate dieser Arbeit sollen nun für den Vergleich der Methoden herangezogen werden.

Datenbasis bilden Elektroenzephalogramm (EEG)-Aufzeichnungen des menschlichen Schlafes. Die Aufzeichnungen sind in einzelne Zeitabschnitte unterteilt und durch Experten den Schlafstadien zugeordnet. Die Daten umfassen insgesamt 37 Probanden und beinhalten Aufzeichnungen von sowohl gesunden als auch schlafgestörten Personen. Hierbei dienten 10 Datensätze von vorwiegend gesunden Probanden der Belehrung der Algorithmen. Die Validierungsdaten beinhalteten 27 Datensätze von gesunden und schlafgestörten Probanden. Eine Auflistung der verwendeten Daten und deren Zuordnung in die Schlafstadien ist in Tabelle 1 dargestellt.

Beim EEG handelt es sich um ein sehr komplexes biologisches Signal. Dieses weist viele Eigenschaften auf, die eine Klassifikation erschweren. Das EEG ist einerseits durch kleine Signalamplituden geprägt, wodurch eine hohe Störanfälligkeit einhergeht. Es zeigen sich außerdem starke inter- und intraindividuelle Ausprägungen in Frequenz, Amplitude und Morphologie [3]. So ist das EEG nicht nur von Faktoren wie dem Alter abhängig. Es wird auch durch Medikamenteneinnahme verändert. Es ergeben sich somit äußerst inhomogene Datensätze, die außerdem, bedingt durch die hohe Individualität der Daten, auch „untypische“ Daten beinhalten können. Diese Daten sind zwar nicht fehlerhaft, behindern jedoch die Bildung eines

generalisierenden Klassifikators [4]. Zudem sind die Schlafklassen innerhalb einer Nacht naturbedingt stark ungleichverteilt, dies wird auch aus Tabelle 1 ersichtlich. Eine nicht zu vernachlässigende Fehlerquelle stellen die zugrunde liegenden Expertenbewertungen dar. Diese Expertenbewertungen, welche Grundlage für die automatische Klassifikation sind, können aus einer Vielzahl von Ursachen abweichen [2]. Ursachen sind zum einen die komplexen Bewertungsregeln des Schlafes, welche nicht immer eindeutig anwendbar sind, und zum anderen die Einbeziehung der Erfahrungen des Experten. Zudem ist der Schlaf ein fließender Prozess, wodurch keine harten Stadienübergänge auftreten. Vor allem in Übergangsbereichen zwischen den Schlafstadien können vermehrt ambivalente Klassifikationen auftreten. Jeder dieser Faktoren zeigt die Komplexität der automatischen Schlafklassifikation auf.

Tabelle 1: Auflistung der verwendeten Daten zur Belehrung und Validierung der Algorithmen

	Belehrung		Validierung	
<i>Datensätze</i>	10		27	
<i>Gesamtdaten</i>	29316		26554	
<i>S1</i>	2860	9,8%	3332	12,5%
<i>S2</i>	12288	41,9%	11450	43,1%
<i>S3</i>	2366	8,1%	1346	5,1%
<i>S4</i>	1009	3,4%	1032	3,9%
<i>REM</i>	5846	19,9%	4540	17,1%
<i>Wach</i>	4499	15,4%	4670	17,6%
<i>Mov</i>	448	1,5%	184	0,7%

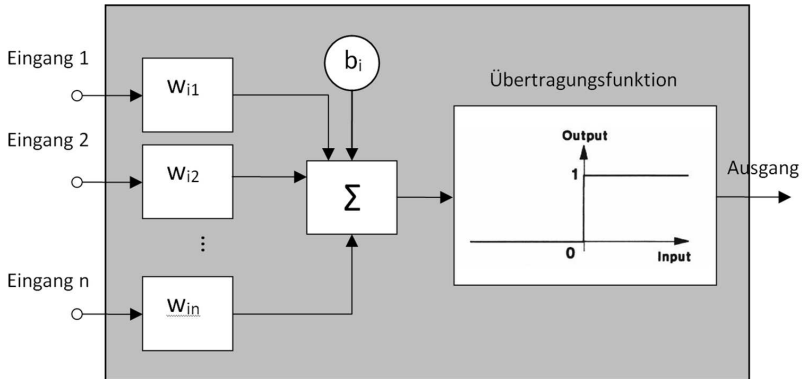


Bild 1: Schematischer Aufbau eines Modellneurons [2]

2 Methoden

2.1 KNN und SVM

Die beiden zu vergleichenden Klassifikationsmethoden unterscheiden sich grundlegend. Die KNN sind nach dem Vorbild des menschlichen Gehirns aufgebaut und bestehen aus miteinander verknüpften einzelnen Neuronen. Jedes Neuron wandelt mehrere gewichtete Eingänge mittels einer Übertragungsfunktion in einen Ausgang um, der wiederum Eingang eines anderen Neurons sein kann. Der Aufbau eines Neurons ist in Bild 1 modellhaft dargestellt.

Hingegen stellt die SVM ein rein mathematisches Verfahren dar, welches versucht zwischen den Punkten zweier Klassen im f -dimensionalen Merkmalsraum eine Trennebene TE mit der Orientierung \underline{w} zu berechnen, die mit einer möglichst maximalen Spannweite M die Klassen voneinander trennt [5][6]. Dies ist in Bild 2 schematisch dargestellt. Aufgrund der methodenbedingten Unterschiede sind Steuermöglichkeiten, Verhalten und Optimierungsmöglichkeiten grundlegend verschieden.

Bei beiden Klassifikationsmethoden wurden eingehende Optimierungsmaßnahmen vorgenommen. So wurden die KNN in [2] in vier Ebenen optimiert. Es erfolgte eine Optimierung der Struktur- und der Belehrungsparameter.

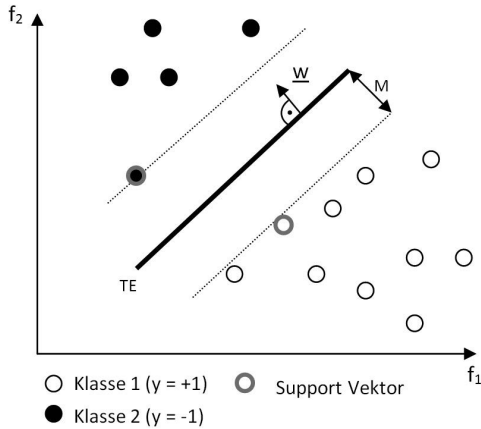


Bild 2: Prinzipielle Funktionsweise der SVM [7]

Außerdem wurde die Optimierung der Merkmalskombinationen in einer dritten Ebene vorgenommen. Die Kooperation der Netzwerk-Populationen mittels eines Zufallssuchverfahrens bildet die vierte Optimierungsebene.

Aus den Untersuchungen in [4] zur SVM-Modellbildung zeigte sich, dass die SVM nur begrenzte Möglichkeiten zur Komplexitätssteuerung besitzt. Außerdem weist das Modell starke Abhängigkeit von den verwendeten Belehrungsdaten auf. So wurde festgestellt, dass bei stark ungleichverteilten Daten, wie sie beim Anwendungsbeispiel der Schlafklassifikation vorliegen, eine Verschiebung der Trennebenen in Richtung der Minoritätsklassen erfolgt und somit die Majoritätsklassen bevorzugt werden [8]. Weiterhin wurde deutlich, dass „untypische“ Daten, Daten die aufgrund ihrer Merkmalsvektoren abseits der Haupt-Datenmenge ihrer Klasse liegen, eine optimale Modellbildung verhindern. So erfolgte zunächst eine Filterung der Daten durch den DiLa-Filter [9] mit anschließender Gleichverteilung. Dieser Filter ermittelt durch Dichte- und Lageinformation aller Daten „untypische“ Daten und filtert diese heraus. Somit können mit den verbleibenden Daten bessere Modelle erzeugt werden. Zudem wurden geeignete Varianten zur Skalierung und eine geeignete Kernelfunktion ermittelt. Eine spezielle Parameteroptimierung mittels dem genetischen Algorithmus SIGA (Special Immigrative multi-criteria Genetic Algorithm) fand Anwendung [10]. Durch die enthaltene neue genetische Operation der Immigration weist der SIGA eine verbesserte Konvergenz auf und trägt zur Steigerung

der Leistungsfähigkeit der Methode bei. Weiterhin konnten die speziellen Eigenschaften dieser Daten für eine kontextbasierte Klassifikation und zur Kombination der Modelle mittels „Bagging“ genutzt werden. Hierfür wurde eine spezielle Ordinalskala für Schlafstadien angewendet.

2.2 Bewertung der Ergebnisse

Zur Bewertung und zum Vergleich der Ergebnisse wurden auf Grundlage der resultierenden Vertauschungsmatrix \mathbf{VM} Gütewerte berechnet. Die Vertauschungsmatrix \mathbf{VM} stellt die absoluten Häufigkeiten der klassifizierten Daten nach Soll- und Ist-Klassifikation dar. Dabei entsprechen die Elemente der Diagonalen von \mathbf{VM} , wo Soll- und Ist-Klassen übereinstimmen, den absoluten Häufigkeiten der richtig klassifizierten Daten r_i . Die Güte eines Klassifikators setzt sich aus mehreren Teilaspekten zusammen. Natürlich sind die relativen Häufigkeiten der richtig klassifizierten Validierungsdaten r_i aller Klassen i ein wichtiger Faktor. Dieser wird in der *Gesamt-Klassifikationsgüte* (GG) mit:

$$GG = 100\% \cdot \frac{\sum_{i=1}^U r_i}{\sum_{i=1}^U g_i} = 100\% \cdot \frac{\sum_{i=1}^U r_i}{N} \quad (1)$$

erfasst, wobei außerdem U die Anzahl der Klassen, N die Gesamtanzahl der Daten und g_i die Gesamtanzahl der Daten einer Klasse i darstellen. Durch die starke Ungleichverteilung der Klassen kann dieser Gütewert aber nicht alleine Aussage über die Qualität eines Klassifikators geben. Es sollen außerdem alle Klassen gut klassifiziert werden. Dies drückt die *Durchschnittliche Klassen-Klassifikationsgüte* (DG) aus.

$$DG = 100\% \cdot \frac{1}{U} \sum_{i=1}^U \frac{r_i}{g_i} \quad (2)$$

Bei vielen Klassifikationsmethoden werden die in der Belehrungsdatenmenge schwach vertretenden Klassen gegenüber der stark vertretenden Klasse benachteiligt. Daher soll die Klassifikationsleistung der schwach vertretenden Klassen mit dem Gütewert *Minimale Klassen-Klassifikationsgüte* (MG) bewertet werden.

$$MG = 100\% \cdot \min_{\forall i} \frac{r_i}{g_i} \quad (3)$$

Weiterhin wird mit Hilfe der Vertauschungsmatrix \mathbf{VM} und einer klassifikationsspezifischen Bewertungsmatrix \mathbf{BM} der *gewichtete Kappa-Koeffizient* κ_w berechnet. Dieser berechnet sich aus:

$$\kappa_w = \frac{p_{0w} - p_{ew}}{1 - p_{ew}} \quad (4)$$

wobei sich die gewichteten relativen Häufigkeiten der richtig klassifizierten Daten p_{0w} aus:

$$p_{0w} = \sum_{i=1}^U \sum_{j=1}^U bm_{ij} \cdot vrel_{ij} \quad (5)$$

und die gewichteten relativen Häufigkeiten der zufällig richtig klassifizierten Daten p_{ew} aus:

$$p_{ew} = \sum_{i=1}^U \sum_{j=1}^U \left(bm_{ij} \sum_{l=1}^U vrel_{il} \sum_{l=1}^U vrel_{lj} \right) \quad (6)$$

ergeben. Hierbei sind bm_{ij} die Elemente der Bewertungsmatrix \mathbf{BM} und $vrel_{ij}$ die Elemente der Vertauschungsmatrix \mathbf{VM} relativ zur Gesamtanzahl N .

Das Gütekriterium des gewichteten Kappa-Koeffizienten wurde von [11] übernommen. Die schlafspezifische Bewertungsmatrix von [2]. Der Wert ist Maximal $\kappa=1$, wenn nur konkordante Klassifikationen auftreten. Wenn die Klassifikation der Zufallswahrscheinlichkeit entspricht, nimmt der gewichtete Kappa-Koeffizient den Wert $\kappa=0$ an. Ein Wert von $\kappa=-1$ bedeutet, dass keine konkordanten Klassifikationen auftreten obwohl die Wahrscheinlichkeit dafür maximal ist. Die Klassifikation ist somit schlechter als der Zufall.

3 Ergebnisse

3.1 Klassifikation von Hypnogrammen

Zunächst sollen die aus der Klassifikation resultierenden Hypnogramme eines Probandendatensatzes, wie in Bild 3 dargestellt, verglichen werden. Dabei ist das Hypnogramm durch die SVM-Klassifikation links und das Hypnogramm durch die optimierten Populationen von KNN rechts dargestellt, die Expertenbewertung ist in beiden Hypnogrammen in hellem Grau hinterlegt.

Zunächst fallen die häufigen Klassenwechsel im Hypnogramm der SVM-Klassifikation auf. Der prinzipielle Verlauf ist zwar erkennbar, jedoch wirken sich diese Klassenwechsel störend aus. Dem entgegen weist die Klassifikation durch KNN klare Verläufe auf. Besonders deutlich wird dies in Zeitabschnitt zwischen 280 und 350 Minuten.

In diesem Zeitabschnitt ist noch ein weiterer Unterschied zwischen den Methoden erkennbar. Während die SVM-Klassifikation die Soll-Klasse REM nur mit den beiden ähnlichsten Klassen, Wach und S1, vertauscht, finden bei der Klassifikation mit KNN auch Vertauschungen mit S2 statt.

Bei dem Hypnogramm durch KNN treten lange fehlklassifizierte Abschnitte auf. Hingegen kann die SVM-Klassifikation durch die starken Schwankungen einige Epochen richtig zuordnen. Dies zeigt sich besonders in den Zeitabschnitten 95-105 Minuten, 180-200 Minuten und 440-465 Minuten. Aufgrund von Optimierungsmaßnahmen können die schweren Vertauschungen durch die SVM-Klassifikation stark reduziert werden. Die Klassifikation mit den KNN zeigt hingegen noch einzelne schwere Vertauschung, wie bei 135 Minuten, wo die Soll-Klasse S2 mit dem Tiefschlafstadium S4 vertauscht wird.

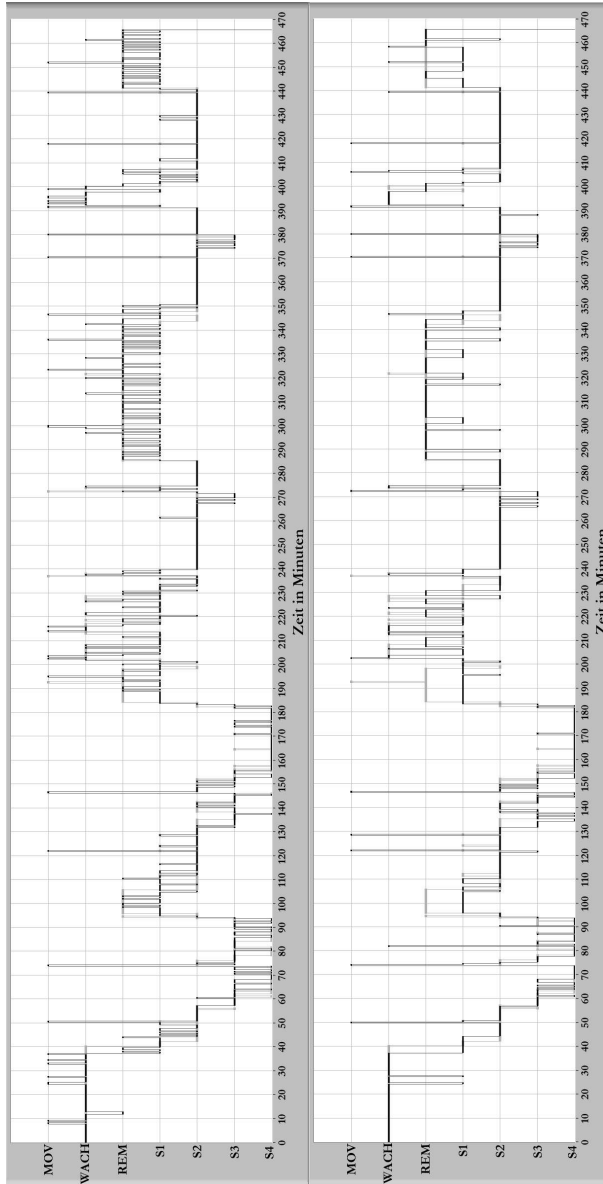


Bild 3: Hypnogramm eines Probanden klassifiziert durch SVM (links) und durch KNN nach [2] (rechts)

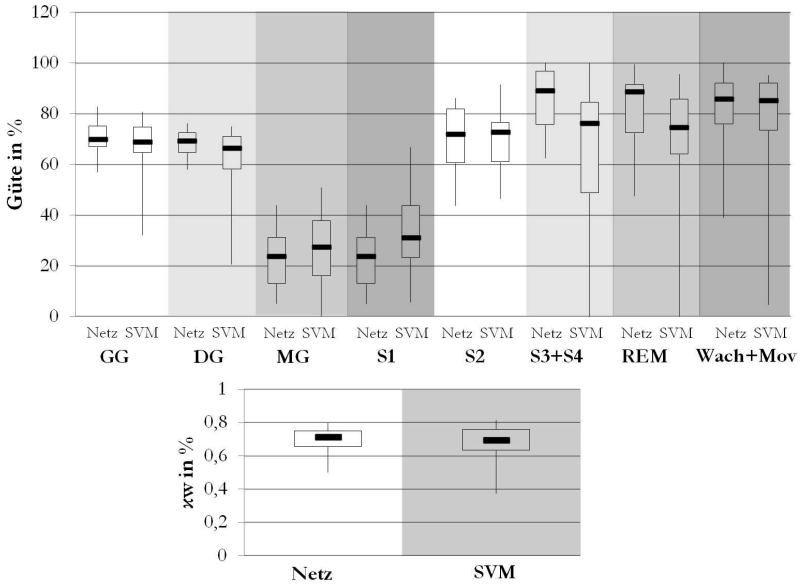


Bild 4: Kastendiagramme des Gütevergleichs KNN [2] und SVM-Klassifikation

3.2 Klassifikation der gesamten Daten

Ein direkter Vergleich der Klassifikationsresultate aller Datensätze ist in Bild 4 in Form von Kastendiagrammen dargestellt. Die Tabelle 2 zeigt die zugehörigen Signifikanzwerte aus dem zweiseitigen Wilcoxon-Vorzeichen-Rang-Test. Die Ergebnisse zeigen die Klassifikationen der gleichen Datensätze durch beide Methoden, wobei die Klassen S3 und S4, sowie Wach und Movement, zusammengefasst wurden. Aus diesen Ergebnissen ist eine höhere Streuung der meisten betrachteten Kriterien bei der SVM-Klassifikation zu erkennen. Dies zeigt, dass bei der SVM-Klassifikation stark abweichende Klassifikationsgütewerte für jeden Datensatz auftreten. Im Gegensatz dazu zeigen die KNN deutlich stabilere Ergebnisse. Außerdem sind die Mediane der meisten betrachteten Kriterien bei den KNN höher. Besonders deutlich wird dies bei den Klassenklassifikationsraten der Klassen S3/S4 und REM. Alle Ergebnisse unterscheiden sich signifikant, wie die Tabelle 2 zeigt. Hierbei liegen die Signifikanzwerte aller Gütewerte unter einem Signifikanzniveau von $\alpha=0,05$.

Tabelle 2: Signifikanzwerte für den Vergleich der Klassifikationsmethoden aus dem zweiseitigen Wilcoxon-Rang-Test

Gütwert	Signifikanzwert
<i>GG</i>	0,0007
<i>DG</i>	0,0001
<i>MG</i>	0,0003
<i>S1</i>	< 0,0001
<i>S2</i>	0,0049
<i>S3 + S4</i>	< 0,0001
<i>REM</i>	0,0014
<i>Wach + Mov</i>	0,0005
<i>κw</i>	0,0006

Der gewichtete Kappa-Koeffizient stellt eine Gesamtgüte der Klassifikation dar, welche auch den Grad der Vertauschungen berücksichtigt. Der Vergleich des gewichteten Kappa-Koeffizienten zeigt nochmals den Unterschied der Klassifikationsleistung beider Methoden. Die SVM-Klassifikation weist im Gegensatz zu der Klassifikation mit den KNN eine höher Streuung auf und zeigt gleichzeitig einen geringeren Median der Resultate.

4 Schlussfolgerungen

Die zuvor aufgestellte These, die SVM sei den KNN überlegen, kann in diesem Beitrag anhand komplexer biologischer Daten nicht bestätigt werden. Obwohl die Klassifikation der Hypnogramme auch einige Vorteile der SVM-Klassifikation gegenüber der KNN aufdeckt, ist die SVM-Klassifikation vor allem durch häufige Klassenwechsel geprägt. Dadurch wird das Hypnogramm insgesamt undeutlicher. Der Vergleich der Klassifikationsgütwerte der gesamten Klassifikation zeigt weiterhin eine instabilere Klassifikation durch die SVM. Dies wird deutlich durch eine hohe Streuung der Ergebnisse. Zudem ergaben die Auswertungen einen geringeren Median für die SVM-Klassifikation in den meisten Gütekriterien.

Es zeigte sich außerdem, dass eine Steuerung der Komplexität des SVM-Modells durch die SVM-Parameter nur eingeschränkt möglich ist [4]. Die Komplexität wird u.a. maßgeblich durch die Belehrungsdaten beeinflusst.

Zum einen beeinflusst die Verteilung der Daten auf die einzelnen Klassen, die Ausbalancierung, die Modellbildung. Zum anderen übt auch die Verteilung innerhalb einer Klasse, speziell die untypischen Daten einen Einfluss aus. Dadurch ist eine Analyse und Bearbeitung der Belehrungsdaten notwendig.

Literatur

- [1] B. Schölkopf und A.J. Smola. „Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond“. MIT Press. 2001.
- [2] A. Wenzel. „Robuste Klassifikation von EEG-Daten durch Neuronale Netze“. Aachen: Shaker Verlag. 2005.
- [3] A. Ebner und G. Deuschl. „EEG“. Stuttgart: Georg Thieme Verlag KG. 2011.
- [4] M. Trommer, M. Schneider und C. Walther. „Auswirkungen von ungleichverteilten und ungenauen Belehrungsdaten auf die Klassifikation der Support Vektor Maschine“. In: *Tag der Forschung 2014*, S. 35–52. Schmalkalden: FHS-prints. 2014.
- [5] V. Vapnik und A. Chervonenkis. „Theorie der Zeichenerkennung“. Berlin: Akademie-Verlag. 1979.
- [6] L. Wang. „Support Vector Machines: Theory and Applications“. New York: Springer. 2005.
- [7] J.A.K. Suykens. „Least Square support vector machines“. Singapore: World Scientific. 2002.
- [8] R. Akbani, S. Kwek und N. Jakowicz. „Applying support vector machines to imbalanced datasets“. In: *Proceedings of European Conference on Machine Learning*, S. 39—50. 2004.
- [9] M. Trommer und A. Wenzel. „Automatische Identifikation und Selektion von untypischen Daten“. In: *at-Automatisierungstechnik 64*, Nr.1. S. 19–28. 2016.

- [10] M. Krautwald, D. Trommer, R. Baumgart-Schmitt, C. Walther und C. Menz. „Parameter optimization for Support Vector Machines by using a multicriteria genetic algorithm for classification of sleep-stages“. In: *Proc., 55th IWK*, S. 527–532. TU Ilmenau. 2010.
- [11] J. Bortz, G.A. Lienert und K. Boehnke. „Verteilungsfreie Methoden in der Biostatistik“. Heidelberg, New York: Springer. 2000.

Classification of Production Quality in Injection Moulding with an Embedded Diagnostic System Using a Fuzzy Inference System

Manuel Schneider¹, Christian Walther^{1,2}, Andreas Wenzel^{1,2}

¹Embedded Diagnostic Systems, Faculty of Electrical Engineering,
Schmalkalden University of Applied Sciences
Blechhammer 9, D-98574 Schmalkalden, Germany
E-Mail: m.schneider@hs-sm.de

²Fraunhofer Institute of Optronics, System Technologies and Image
Exploitation, Advanced System Technologies
Am Vogelherd 50, D-98693 Ilmenau, Germany
E-Mail: christian.walther@iosb-ast.fraunhofer.de,
andreas.wenzel@iosb-ast.fraunhofer.de

Abstract

The construction of a fuzzy inference system for calculation of the production quality within injection moulding processes is described in this contribution. The aim of this work is to get an automatic decision for the quality of the production cycle. For the measurement pressure sensors are used within the cavity of the moulding tool. The sensor signals were sampled and analysed by an embedded diagnostic system that was integrated into the injection moulding tool.

1 Introduction

Plastic products have many different application fields e.g. in automotive, medicine, aeronautics and many more. The efficiency of an injection moulding tool depends on the number optimal made products. The regulation and the classification of the production quality is made by the injection moulding machine itself. As an extension to this existing technique a so-called „Powermould“-tool was developed. The outline of this „Powermould“-tool was presented in [1]. This „Powermould“-tool describes

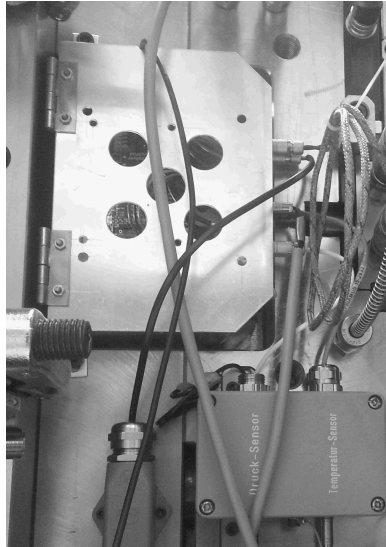


Figure 1: Photograph of the measurement hardware (EDS) integrated into the injection moulding tool.

a mould tool with special equipment, sensors and skills. The underlying research project is an undertaking of the Thuringian Centre of Mechanical Engineering (THZM⁶). The aim of this project is to implement an intelligent embedded diagnostic system (EDS) into a mould tool. This EDS could be used independently at the injection moulding machine and supports the machine operator to get faster the optimal working point for the production cycle. For measurement purposes four pressure, six temperature and two force sensors were implemented. For this article only one pressure sensor was used. In the figure 1 a photograph of the EDS is displayed. The photograph also shows one side of the used injection moulding tool at which the EDS is mounted.

⁶THZM - German abbreviation for Thuringian centre of mechanical engineering (Thüringer Zentrum für Maschinenbau)

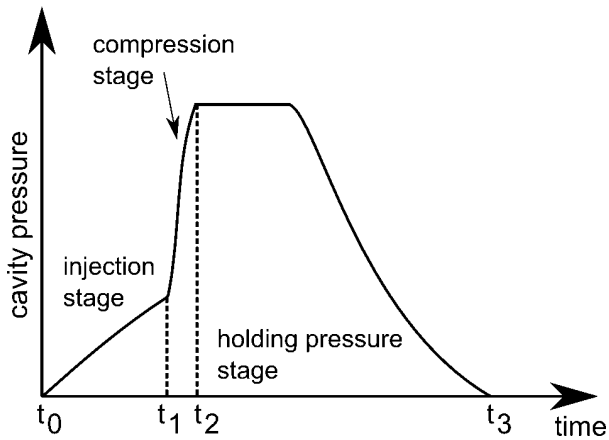


Figure 2: Change in cavity pressure over time and the different production stages [2].

2 Methods

The determination of the optimal working point is necessary for every injection moulding tool. This working point depends on melt temperature, mould temperature, injection time, compression time and the holding pressure time. This process of adjustment is the basis for a good production quality. The figure 2 shows an outline of an optimal pressure curve as presented in [2].

In a first step a large number of production cycles was admitted. For the experiment a two-stage fractional factorial experimental design with six factors and four central points was created. The results of this design of experiment was described in [4] and [5]. Within a first investigation a decision trees were used to classify the production quality [3]. The features presented in [3] were used for generation a fuzzy inference system. To describe the characteristics of the pressure signals 16 features were extracted within a first approach of analysis. This 16 features were split into three categories to describe the different stages of the production cycle. The first important point in the pressure curve is the switch-over point from compression stage to the holding pressure stage (see t_2 in figure 2). The second one is the height of the cavity pressure during the holding pressure time. The last important stage is the falling behaviour during the holding

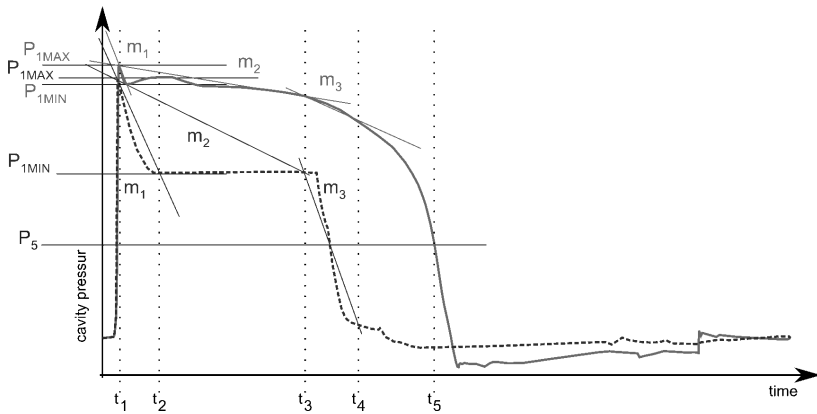


Figure 3: Structural characteristics of an optimal (solid line) and middle (stroked line) pressure signal measured by EDS[6].

pressure time. These three parts of the production cycle determine the production quality of the product. Examples of two different production cycles were presented in figure 3 in [6]. The most of this features are describing structural characteristics of the measured signal. The solid graph in figure 3 shows a optimal pressure signal. The stroked graph is an example for a poor production cycle. As you can see the pressure height at the holding pressure stage is to low and the time for this stage is to short. But in comparison to figure 2 the pressure signals from figure 3 of the measurements are quite similar. The reason for the difference between this charts could be the developed chopper amplifier. This amplifier was specially developed for the piezo electrical pressure sensors. In details it was described in [7], [8] and [9].

By using a design of experiments 651 production cycles were measured and analysed. An injection moulding expert has rated the measured pressure signals into three classes: *poor* (386 datasets), *average* (174 datasets) and *excellent* (91 datasets). The expert or machine operator rated the production cycles by the information of the pressure signal and the visual quality of the produced object. In a first step the expert assigned grades to the three stages of the production cycle. The summation of this three grades yields into the result of the production quality classes. The table 1 shows some examples of the rating of the production cycles by the machine operator. The explanation of the numbers used in table 1 is shown in table 2.

Table 1: Examples of the result from the expert rating of the three production stages. The meaning of this numbers is listed in table 2.

production cycle name	switchover point	holding pressure	holding pressure time	result
01_02	2	1	1	4
02_01	3	1	1	5
04_08	9	9	9	27
13_08	1	1	1	3
16_04	3	1	2	6
⋮	⋮	⋮	⋮	⋮

A dataset of one production cycle consists of the calculated 16 feature values and the result of the expert rating. For the establishment and validation of the fuzzy inference system the collected data were split into 455 training datasets and 196 testing datasets. The dataset of one class was split in 70 % training and 30 % validation quantities. To calculate the fuzzy inference system (fis) MATLAB[®] was used. At the beginning the training dataset was used to determine the input fis Sugeno type fuzzy structure. For this calculation the genfis2-function was used. The genfis2-function requires one structure of input datasets (16 features) and one structure of a output dataset (result of the expert rating). The genfis2-function extract the rules with a clustering procedure and determines the number of rules and the membership functions. The linguistic variables was calculated from every input dataset by the algorithm itself. The next step within genfis2-function the least square estimation is used to calculate each rule's consequent equation and save it in a fis structure.

Table 2: Explanation of the numbers used in table 1 to rate the three production stages.

switchover point	holding pressure	holding pressure time	result
1 = optimal	1 = optimal	1 = enough	3 - 4 = excellent
2 = to early	2 = to low	2 = to short	5 - 6 = average
3 = to late	3 = to high	9 = n.e.	>7 = poor
9 = not evaluable (n.e.)	4 = solidified melt 9 = n.e.		

3 Results

The figure 4 shows the result of the training fis structure from the genfis2 algorithm. The classification result of the machine operator is marked as circles and the results of the genfis2-function structure as crosses.

This fis structure was one of the function inputs for the anfis-function. The anfis-function uses a hybrid learning process to tune the Sugeno fis structure. The training dataset is only used for the genfis2 and the anfis-function. For the validation of the fuzzy inference system the evalfis-function was used. The validation datasets are not used for training and testing. The estimated fis structure and the validation dataset were the input arguments for the evalfis-function. To estimate a classification notation, two mathematics operation were needed. The results need to be rounded to the next integer value and the classification values needs to be limited. The result of the evalfis-function by using all 16 features and after 200 epochs of the anfis-function is displayed in figure 5.

The figure 5 shows the anfis-function result as crosses and the expert classification as circles using the validation dataset. The notation is 3 = excellent, 2 = average and 1 poor. The table 3 shows the concordances of the evalfis-function results to the expert classification results.

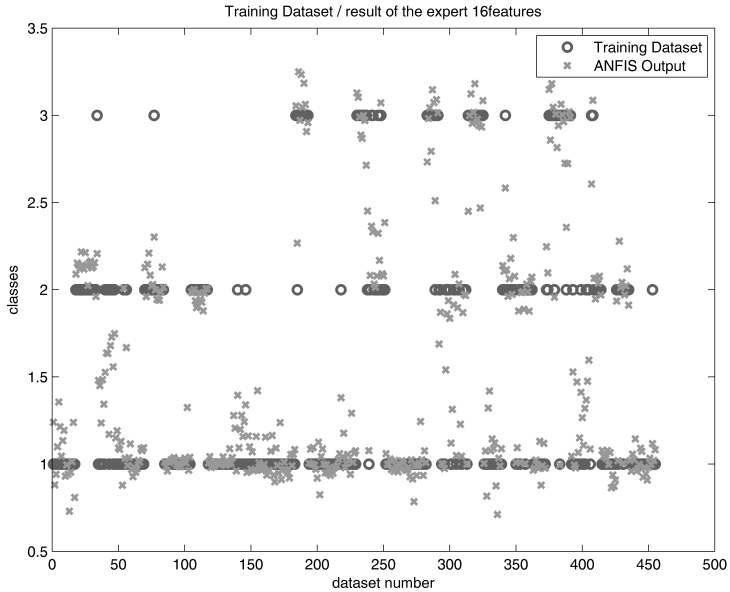


Figure 4: Plot of the training result of the genfis2-function (crosses) and the expert classification (circles) 3 = excellent, 2 = average and 1 poor

In coordination with the expert the numbers of features were reduced from 16 to 7. The 7 features are representing also the three production stages and have good characteristics for the classification. The classification result of the evalfis-function within 7 features is displayed in figure 6 and table 4. For this calculation also 200 training epochs for the anfis algorithm were used.

4 Conclusion

In this contribution the construction of a Takagi Sugeno fuzzy inference system was presented. It was established for the automatic classification of the production quality for an injection moulding process. The optimal results of the fuzzy inference system were created by the combination of 7 features which was presented in table 4. With the reduced feature space the calculation time for the anfis-function was decreased to approximately 33 %.

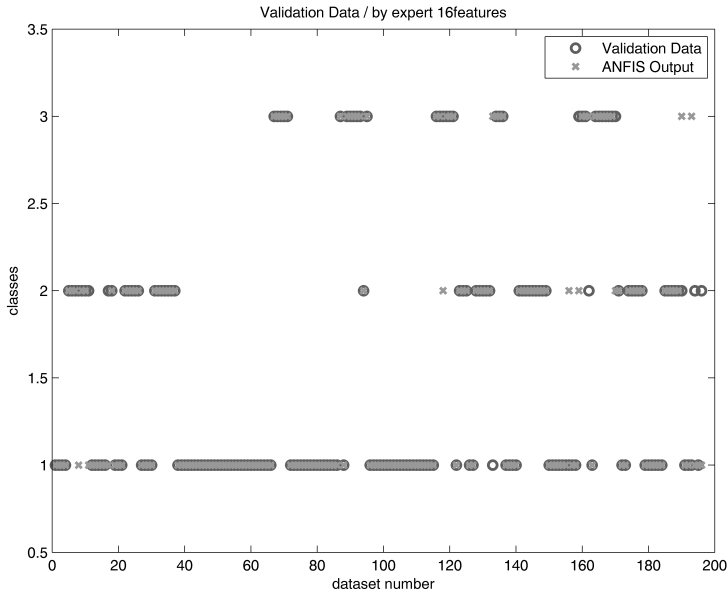


Figure 5: Evalfis result of the validation dataset represent as three class notation by using 16 features for the fis structure

It was presented that the fuzzy inference system can be used to classify the production quality. In future work the fuzzy inference system will be implemented into the EDS to give the machine operator some information about the production quality without an external computer.

Acknowledgements

Special thanks to the Thüringer Aufbaubank as a project sponsor for this research project. The results of the underlying work were funded by the State of thuringia and the European Union (ESF), project number 2013 FRG 0130. The responsibility for the content of this publication lies with the authors.

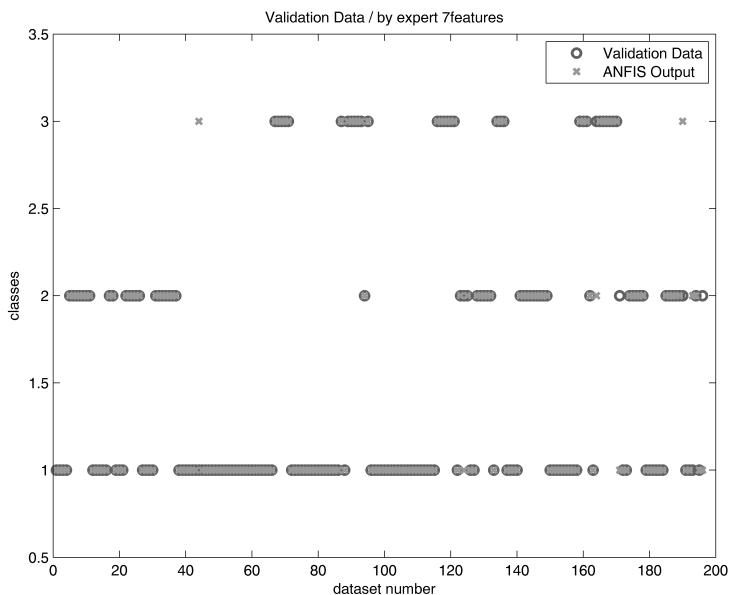


Figure 6: Evalfis result of the validation dataset represent as three class notation by using 7 features for the fis structure

Table 3: Concordances (after 200 training epochs) between the evalfis classification and the expert meaning by using all 16 features and the validation dataset.

	fis	excellent	average	poor
expert				
excellent		90,32 %	9,68 %	0,00
average		3,70 %	87,04 %	9,26 %
poor		1,80 %	0,90 %	97,30 %

Table 4: Concordances (after 200 training epochs) between the evalfis classification and the expert meaning by using a reduced feature space with 7 features and the validation dataset.

	fis	excellent	average	poor
expert				
excellent		96,77 %	3,23 %	0,00
average		1,85 %	92,59 %	5,56 %
poor		0,90 %	0,90 %	98,20 %

References

- [1] M. Schneider and A. Wenzel: „Entwurf eines Eingebetteten Diagnose-systems zur Überwachung von Prozessparametern bei Spritzgießen“, Schmalkalden University of Applied Sciences, *Tag der Forschung*, DOI: 10.5162/sensoren2016/P9.2, 2016.
- [2] F. Johannaber: „Injection Molding Machines A User’s Guide“, In: Carl Hanser Verlag, 4. Auflage, 10/2007, ISBN: 978-3-446-22581-7.
- [3] M. Schneider, R. Jahn and A. Wenzel: „Erprobung eines echtzeitfähigen Auswertungsalgorithmus zu Bewertung der Fertigungsqualität beim Spritzgießen mit Hilfe eines eingebetteten Diagnosesystems“, *17. Nachwuchswissenschaftlerkonferenz*, P 317–323, DOI: 10.5162/sensoren2016/P9.2, 2016.
- [4] T. Seul, A. Wenzel, M. Schneider, P. Röstel, R. Jahn and R. Schlutter: „Auf die inneren Werte kommt es an.“, In: *Kunststoffe 11/2015*, P 46 - 50.
- [5] T. Seul, A. Wenzel, M. Schneider, P. Röstel, R. Jahn and R. Schlutter: „It’s the inner Values that Count.“, In: *Kunststoffe International 11/2015*, P 17 - 21.
- [6] M. Schneider, R. Jahn and A. Wenzel: „(Poster) Entwicklung eines unipolaren differentiellen Ladungsverstärkers für die Anwendung in eingebetteten Diagnosesystemen zur Druckmessung in Spritzgussmaschinen“, *17. Nachwuchswissenschaftlerkonferenz*, DOI: 10.13140/RG.2.1.1042.9849, 2016.
- [7] M. Schneider, A. Jahn, N. Greifzu and N. Fränzel: „Entwicklung eines unipolaren differentiellen Ladungsverstärkers für die Anwendung in eingebetteten Diagnosesystemen zur Druckmessung in Spritzgussmaschinen“, *18. GMA/ITG-Fachtagung Sensoren und Messsysteme*, P 782–789, DOI: 10.5162/sensoren2016/P9.2, 2016.
- [8] N. Greifzu, Master’s thesis: „Entwicklung von Hard- und Software zur Messung von Kraft-, Druck- und Temperatursignalen in Kunststoffspritzgussmaschinen“, Schmalkalden University of Applied Sciences, 2015
- [9] A. Jahn. Bachelor’s thesis: „Konzeption, Umsetzung und Test einer analogen Signalaufbereitungsschaltung für piezoelektrische Drucksensoren“, Schmalkalden University of Applied Sciences, 2015.

Entwurf und Vergleich nichtlinearer Beobachter am realen Viertanksystem, basierend auf Takagi-Sugeno-Modellen

Christian Dengler

Technische Universität München
Boltzmannstr. 15, D-85748 Garching bei München
E-Mail: c.dengler@tum.de

1 Einführung

Die Synthese linearer und nichtlinearer Beobachter gehört zu den klassischen Problemen der Regelungstechnik und findet sowohl in der Industrie als auch in der Forschung häufig Anwendung. Insbesondere bei nichtlinearen Beobachtern fehlen oft Konvergenzbeweise oder die Parameter der Beobachter sind schwer einzustellen. Bei den hier vorgestellten nichtlinearen Beobachtern, welche auf Takagi-Sugeno-Systemen basieren, kommen hingegen auf den ersten Blick wenig Freiheitsgrade vor und im Entwurf ist der Konvergenzbeweis über eine quadratische Lyapunov-Funktion enthalten.

In diesem Beitrag werden zum einen die Auswirkungen der Takagi-Sugeno-Modellierung als zusätzlichen Freiheitsgrad auf einen nichtlinearen Beobachterentwurf untersucht. Es wird untersucht inwieweit nicht nur die Wahl des Beobachter-Entwurfsverfahrens und der gegebenenfalls freien Parameter, sondern auch die Wahl der linearen Teilsysteme des Takagi-Sugeno-Modells Auswirkungen auf die erreichte Beobachterdynamik haben. Zusätzlich zu den zwei vorgestellten Modellierungsmethoden für Takagi-Sugeno-Systeme werden zwei Entwurfsverfahren, der Thau-Luenberger-Beobachter und der Sliding-Mode-Beobachter, verglichen, sodass insgesamt vier unterschiedliche Beobachter im Vergleich stehen. Der Vergleich hinsichtlich bestimmter Kriterien findet sowohl in der Simulation als auch am realen Viertanksystem statt.

Für den Konvergenzbeweis der vorgestellten Beobachter muss eine a priori Annahme getroffen werden, welche in den meisten Arbeiten nach dem Entwurf nicht weiter betrachtet wird. Deshalb wird zusätzlich auf das Einhalten der Annahme eingegangen.

2 Takagi-Sugeno-Modellierung

Die betrachtete Systemklasse der Takagi-Sugeno(TS)-Systeme lautet

$$\begin{aligned}\dot{\mathbf{x}} &= \sum_{i=1}^N h_i(\mathbf{x}, \mathbf{u}) \cdot (\mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u} + \mathbf{d}_i) \\ \mathbf{y} &= \mathbf{C} \mathbf{x},\end{aligned}\tag{1}$$

wobei $h_i(\mathbf{x}, \mathbf{u})$ nichtlineare Gewichtungs- oder Zugehörigkeitsfunktionen sind, welche die Eigenschaften

$$\begin{aligned}h_i(\mathbf{x}, \mathbf{u}) &\in [0, 1], \quad \forall \mathbf{x}, \mathbf{u}, i \\ \sum_{i=1}^N h_i(\mathbf{x}, \mathbf{u}) &= 1, \quad \forall \mathbf{x}, \mathbf{u}.\end{aligned}\tag{2}$$

erfüllen, und N die Anzahl an Teilsystemen angibt.

Um ein System in TS-Darstellung zu erhalten, gibt es drei unterschiedliche Vorgehensweisen. Zum einen kann ein TS-System als Black-Box-Ansatz für eine Systemidentifikation angesetzt werden. Dieser Ansatz wird im folgenden nicht weiter betrachtet, da für beobachtete Strecken für gewöhnlich nicht alle Sensoren vorhanden sind um die Identifikation durchzuführen. Die beiden anderen Vorgehensweisen gehen von einer nichtlinearen Funktion

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})\tag{3}$$

aus. Entweder kann das System nun durch Linearisierungen approximiert, oder durch Umformen exakt dargestellt werden. Die Methode der exakten Umformung ist in der Literatur als Sektor-Nichtlinearitäten-Ansatz bekannt. Die beiden letzteren Vorgehensweisen um auf die TS-Struktur (1) zu gelangen werden in diesem Beitrag untersucht und im folgenden kurz erläutert.

2.1 Sektor-Nichtlinearitäten-Ansatz

Für die Erstellung eines TS-Modells über eine Umformung mittels des Sektor-Nichtlinearitäten-Ansatzes muss zuerst der Gültigkeitsbereich $\mathbf{\Gamma}$

festgelegt werden. Nur in diesem Gültigkeitsbereich werden die Gewichtungsfunktionen $h_i(\mathbf{x}, \mathbf{u})$, wie sie aus diesem Ansatz entstehen, die Eigenschaften (2) erfüllen. Nachdem $\mathbf{\Gamma}$ festgelegt wurde, wird jede Nichtlinearität umgeformt so, dass ein Produkt aus einer nichtlinearen Funktion $g_j(\mathbf{x}, \mathbf{u})$ und einer Zustands- oder Eingangsgröße entsteht, zum Beispiel wird \sqrt{x} umgeformt zu $\frac{\sqrt{x}}{x}x$. Hierbei ist $j \in \{1, \dots, M\}$ mit M der Anzahl an auftretenden Nichtlinearitäten. Danach wird für jede Nichtlinearität das Maximum und Minimum im Gültigkeitsbereich $\mathbf{\Gamma}$ gesucht und zwei Hilfsfunktionen $w_{j,\min}(\mathbf{x}, \mathbf{u}) = \frac{g_{j,\max} - g_j(\mathbf{x}, \mathbf{u})}{g_{j,\max} - g_{j,\min}}$, $w_{j,\max}(\mathbf{x}, \mathbf{u}) = \frac{g_j(\mathbf{x}, \mathbf{u}) - g_{j,\min}}{g_{j,\max} - g_{j,\min}}$ erstellt. Die nichtlinearen Gewichtungsfunktionen ergeben sich aus den möglichen Kombinationen von

$$h_i(\mathbf{x}, \mathbf{u}) = \prod_{j=1}^M w_{j,\min/\max}(\mathbf{x}, \mathbf{u}), \quad i \in \{1, \dots, 2^M\}. \quad (4)$$

Die zugehörigen linearen Teilsysteme ergeben sich durch Ersetzen von allen $g_j(\mathbf{x}, \mathbf{u})$ in $\mathbf{f}(\mathbf{x}, \mathbf{u})$ durch $g_{j,\min}$ falls $w_{j,\min}(\mathbf{x}, \mathbf{u})$ im Produkt (4) verwendet wurde, oder durch $g_{j,\max}$ im Falle von $w_{j,\max}(\mathbf{x}, \mathbf{u})$. Es ergeben sich somit immer 2^M nichtlineare Teilsysteme. Ausführliche Beispiele finden sich z.B. in [7, S. 10-23].

2.2 Approximation über Linearisierung

Wie schon beim Sektor-Nichtlinearitäten-Ansatz wird zuerst ein kompakter Raum $\mathbf{\Gamma} \subset \mathbb{R}^n$ als Gültigkeitsbereich für das zu entwerfende TS-Modell festgelegt. In diesem Raum werden nun Linearisierungspunkte $\mathbf{x}_{\text{lin},i}, \mathbf{u}_{\text{lin},i}$ ausgewählt, und die Teilsysteme an den entsprechenden Punkte über die Taylor-Entwicklung berechnet als

$$\begin{aligned} \mathbf{A}_i &= \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}^T} \right|_{\mathbf{x}_{\text{lin},i}, \mathbf{u}_{\text{lin},i}}, \quad \mathbf{B}_i = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}^T} \right|_{\mathbf{x}_{\text{lin},i}, \mathbf{u}_{\text{lin},i}} \\ \mathbf{d}_i &= \mathbf{f}(\mathbf{x}_{\text{lin},i}, \mathbf{u}_{\text{lin},i}) - \mathbf{A}_i \mathbf{x}_{\text{lin},i} - \mathbf{B}_i \mathbf{u}_{\text{lin},i}. \end{aligned} \quad (5)$$

Die Wahl des Gitters an Linearisierungspunkten in $\mathbf{\Gamma}$ in ist hierbei entscheidend für eine gute Approximation mit möglichst wenige Teilsysteme. Abschließend werden Zugehörigkeitsfunktionen vorgegeben, welche die Bedingungen (2) erfüllen, z.B. Dreiecksfunktionen, Trapezfunktionen, normierte Gauß-Glocken, oder Andere.



Bild 1: Das Viertanksystem am Lehrstuhl für Regelungstechnik der TU München.

3 Das Viertanksystems

Das Viertanksystem (siehe Bild 1) besteht aus zwei Pumpen, welche über Schläuche mit vier Wassersäulen und zwei Auffangbecken verbunden sind. Beim Anlegen einer entsprechenden Spannung an die Pumpen wird Wasser aus den Auffangbecken in die Wassersäulen gepumpt. Jede Pumpe versorgt dabei zwei Wassersäulen, welche über Kreuz angeordnet sind. Das Aufteilen des Wasserflusses in die zwei mit der Pumpe verbundenen Wassersäulen ist hierbei über ein Ventil vorgegeben. Das Wasser fließt dann aus der Wassersäule entweder in die darunter befindliche Wassersäule oder zurück in das Auffangbecken. Das Viertanksystem ist ein Vorzeige-System in der Regelungstechnik für verschiedene Konzepte der Mehrgrößenregelung und der nichtlinearen Regelungstechnik [6] und wird hier als Benchmarksystem für nichtlineare Beobachter herangezogen. Hierzu wird angenommen, dass nur die Höhe des Wassers der unteren beiden Wassersäulen gemessen werden kann, während der Wasserstand der oberen Säulen über Beobachter geschätzt wird.

3.1 Modellierung

Für die Modellierung des Viertanksystems werden die Bernoulli-Gleichungen und die Massenbilanz verwendet. Da Wasser als inkompressibel angenommen wird, kann die Massenbilanz für jede der vier Wassersäulen auch als Volumenbilanz umgeschrieben werden als

$$\dot{V}_{\text{sys},i} = \dot{V}_{\text{zu},i} - \dot{V}_{\text{ab},i}. \quad (6)$$

Die Füllstandshöhe der einzelnen Tanks sei x_i , $i \in \{1, 2, 3, 4\}$. Da der Querschnitt A der Wassersäulen konstant ist, kann die Volumenänderung in den Tanks geschrieben werden als

$$A\dot{x}_i = \dot{V}_{\text{zu},i} - \dot{V}_{\text{ab},i} \quad (7)$$

Betrachtet man nun nur den Abfluss aus einer Säule $\dot{V}_{\text{ab},i}$, und sei $v_{\text{ab},i}$ der Volumenstrom am zugehörigen Abfluss mit Querschnittsfläche a , kann man schreiben

$$A\dot{x}_i = av_{\text{ab},i}. \quad (8)$$

Zusammen mit der Bernoulli-Gleichung, aufgestellt an der Wasseroberfläche und des Abflusses der Säule, erhält man

$$v_{\text{ab},i} = A\sqrt{\frac{2g}{A^2 - a^2}}\sqrt{x_i}. \quad (9)$$

Betrachtet man nun nur den Zufluss, so ergibt dieser sich aus dem Volumenstrom aus der Pumpe über das Ventil, und für x_1 und x_2 zusätzlich aus dem Zufluss aus der darüber liegenden Wassersäule. Der Volumenstrom der Pumpe wird proportional zu Pumpenspannung angenommen als $\dot{V}_{\text{Pumpe},i} = k_p u_i$. Die Verteilung des Wasserstroms über die Ventile wird mit $0 < \gamma < 1$ berücksichtigt ergibt sich für die oberen Tanks zu

$$\dot{V}_{\text{Pumpe},\text{zu},1} = (1 - \gamma)k_p u_i \quad (10)$$

und für die unteren Tanks zu

$$\dot{V}_{\text{Pumpe},\text{zu},2} = \gamma k_p u_i. \quad (11)$$

Fügt man die Gleichungen für die jeweiligen Wassersäulen zusammen, erhält man mit

$$\alpha = a\sqrt{\frac{2g}{A^2 - a^2}}$$

Tabelle 1: Parameter des Viertanksystems

Beschreibung	Formelzeichen	Größe
Querschnittsfläche der Säule	A	17.35 cm ²
Querschnittsfläche des Abflusses aus der Säule	a	0.159 cm ²
Pumpenkonstante	k_p	3.3 cm ³ Vs ⁻¹
Gravitationskonstante	g	981 cm s ⁻²
Ventilstellungen	γ	0.4

$$\beta = \frac{k_p}{A}.$$

die Zustandsraumdarstellung

$$\begin{aligned}
 \dot{x}_1(t) &= -\alpha \sqrt{x_1(t)} + \alpha \sqrt{x_3(t)} + \beta \gamma_1 u_1(t) \\
 \dot{x}_2(t) &= -\alpha \sqrt{x_2(t)} + \alpha \sqrt{x_4(t)} + \beta \gamma_2 u_2(t) \\
 \dot{x}_3(t) &= -\alpha \sqrt{x_3(t)} + \beta (1 - \gamma_2) u_2(t) \\
 \dot{x}_4(t) &= -\alpha \sqrt{x_4(t)} + \beta (1 - \gamma_1) u_1(t).
 \end{aligned} \tag{12}$$

Gleichung (12) wird im folgenden als $\dot{\mathbf{x}} = \mathbf{f}_{Vt}(\mathbf{x}, \mathbf{u})$ abgekürzt. Die verwendeten Streckenparameter sind in Tabelle 1 angegeben.

3.2 Exakte TS-Darstellung des Viertanksystems

Da die Systemgleichungen vier Nichtlinearitäten enthalten ($\sqrt{x_1}$, $\sqrt{x_2}$, $\sqrt{x_3}$, $\sqrt{x_4}$), enthält das TS-Modell aus der exakten Umformung $2^4 = 16$ Teilsysteme. Durch die nötige Umformung $\sqrt{x} = \frac{\sqrt{x}}{x} x$ kann der Gültigkeitsbereich den Ursprung nicht beinhalten, und der Gültigkeitsbereich wird $\Gamma_1 = \{\mathbf{x} \in \mathbb{R} | 0.5 \leq x_i \leq 25\}$ gewählt.

3.3 Approximative TS-Darstellung des Viertanksystems

Für die Wahl der Linearisierungspunkte wurde ein Algorithmus geschrieben, welcher basierend auf einer Fehlernorm neue Linearisierungspunkte erstellt. Beginnend beim Punkt $\mathbf{x}_{lin,1} = [0.5 \ 0.5 \ 0.5 \ 0.5]^T$, wird in eine Zustandsrichtung $j = 1$ der nächste Punkt $\mathbf{x}_{lin,2}$ dann erstellt, wenn die Fehlernorm $\|\mathbf{f}_{Vt}(\mathbf{x}_{lin,1} + \delta_{\mathbf{x}} \mathbf{e}_j, \mathbf{0}) - \mathbf{A}_i \mathbf{x}_{lin,1} - \mathbf{d}_i\|$ eine vorgegebene Toleranz

überschreitet. \mathbf{e}_j beschreibt hier ein Vektor der gleichen Länge wie \mathbf{x} , mit einer 1 an Stelle j , und alle anderen Einträge 0. Der Ansatz über die Fehlernorm wird dann mit $\mathbf{x}_{\text{lin},1}$ wiederholt so lange bis der Rand von $\Gamma_2 = \Gamma_1$ erreicht wird. Der Ursprung ist auch hier nicht enthalten, da (12) hier nicht stetig differenzierbar ist, und sonst auch kein fairer Vergleich möglich wäre. Das Vorgehen wird dann für jeden bisher erstellten Linearisierungspunkt in Zustandsrichtung $j = 2$, dann $j = 3$ u.s.w. wiederholt. Die Toleranz wurde 0.25 gewählt, was auf ein Gitter mit insgesamt 144 Linearisierungspunkten führt. Ein Ausschnitt aus der x_1x_2 -Ebene des Gitter ist in Bild 2 zu sehen. Die Zugehörigkeitsfunktionen wurden als

$$w(x, \Theta) = \begin{cases} 1 & , \text{ falls } \Theta_2 \leq x \leq \Theta_3 \\ \frac{1}{2} - \frac{1}{2} \cos\left(\frac{x-\Theta_1}{\Theta_2-\Theta_1}\pi\right) & , \text{ falls } \Theta_1 \leq x < \Theta_2 \\ \frac{1}{2} - \frac{1}{2} \cos\left(\frac{\Theta_4-x}{\Theta_4-\Theta_3}\pi\right) & , \text{ falls } \Theta_3 < x \leq \Theta_4 \\ 0 & , \text{ sonst} \end{cases} \quad (13)$$

$$h_i(\mathbf{x}) = \prod_{j=1}^4 w(x_{\text{lin},i;j}, \Theta_{i,j}) \quad (14)$$

ausgewählt, mit $\Theta_{i,j}$ so, dass die Zugehörigkeitsfunktionen sich jeweils auf einem Drittel des Abstandes zwischen den Linearisierungspunkten überschneiden.

Bild 3 zeigt eine Trajektorie mit den beiden Takagi-Sugeno-Systemen in der Simulation, sowie der zugehörigen Messung am realen System. Beide Takagi-Sugeno-Systeme besitzen eine ähnliche Dynamik, da beide auf der gleichen analytischen Funktion basieren. Zusätzlich anzumerken ist, dass das approximative TS-System aufgrund der Abweichung zur analytischen Funktion eine Ruhelage für $\mathbf{u} = \mathbf{0}$ mit negativen Wasserständen aufweist.

4 Thau-Luenberger-Beobachter

In diesem Abschnitt wird ein Beobachterentwurf basierend auf Takagi-Sugeno-Modellen hergeleitet. Der angegebene Beobachterentwurf wurde in [2] vorgestellt und in [3] auf die hier vorgestellte Form erweitert.

Die Beobachterdynamik ist ähnlich der Systemdynamik

$$\dot{\hat{\mathbf{x}}} = \sum_{i=1}^N h_i(\hat{\mathbf{x}}) (\mathbf{A}_i \hat{\mathbf{x}} + \mathbf{B}_i \mathbf{u} + \mathbf{d}_i + \mathbf{L}_i (\mathbf{y} - \mathbf{C} \hat{\mathbf{x}})) \quad (15)$$

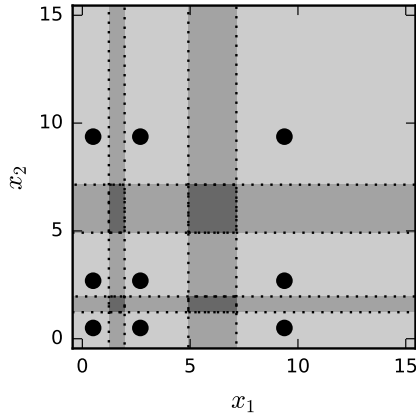


Bild 2: Gitter an Linearisierungspunkten in der x_1x_2 -Ebene, mit Überschneidungsbereiche der Zugehörigkeitsfunktionen.

als TS-System mit zusätzlichem Eingang $\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}$ modelliert, wobei $\hat{\mathbf{x}}$ die beobachteten Zustandsgrößen darstellt. Das Ziel des Entwurfs ist die Bestimmung der Matrizen \mathbf{L}_i so, dass der Beobachterfehler

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} \quad (16)$$

asymptotisch abklingt.

Die Beobachterdynamik ergibt sich durch Einsetzen von (1) und (15) in $\dot{\mathbf{e}} = \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}}$ zu

$$\begin{aligned} \dot{\mathbf{e}} = & \sum_{i=1}^N h_i(\hat{\mathbf{x}}) (\mathbf{A}_i - \mathbf{L}_i\mathbf{C}) \mathbf{e} + \\ & \sum_{i=1}^N (h_i(\mathbf{x}) - h_i(\hat{\mathbf{x}})) (\mathbf{A}_i\mathbf{x} + \mathbf{B}_i\mathbf{u} + \mathbf{d}_i). \end{aligned} \quad (17)$$

Um Stabilität mittels einer quadratischen Lyapunov-Funktion nachweisen zu können muss zusätzlich eine Annahme zum Term

$$\Delta = \sum_{i=1}^N (h_i(\mathbf{x}) - h_i(\hat{\mathbf{x}})) (\mathbf{A}_i\mathbf{x} + \mathbf{B}_i\mathbf{u} + \mathbf{d}_i) \quad (18)$$

getroffen werden. Da

$$\lim_{\hat{\mathbf{x}} \rightarrow \mathbf{x}} \Delta = \mathbf{0} \quad (19)$$

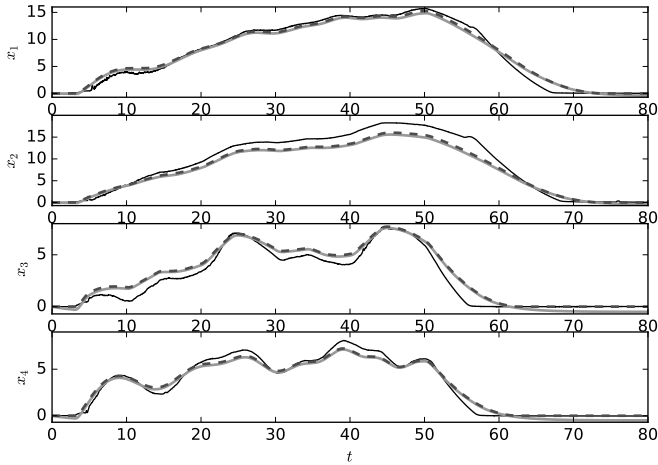


Bild 3: Messung im Vergleich mit einer Simulation der Takagi-Sugeno-Modelle, schwarze Linie: Messung, grau-gestrichelte Linie: TS-Modell aus exakter Umformung, grau-durchgezogene Linie: TS-Modell aus Linearisierungen.

wird Δ als Lipschitz-stetig angenommen, mit Lipschitz-Konstante α :

$$\|\Delta\| \leq \alpha \|\mathbf{x} - \hat{\mathbf{x}}\|. \quad (20)$$

Um das asymptotische Abklingen des Beobachterfehlers zu garantieren, wird die quadratische Lyapunov-Funktion

$$V(\mathbf{e}) = \mathbf{e}^T \mathbf{P} \mathbf{e}, \quad \mathbf{P} > \mathbf{0} \quad (21)$$

angesetzt. Die zeitliche Ableitung der Lyapunov-Funktion muss negativ definit sein. Die Ableitung führt auf

$$\begin{aligned} \dot{V}(\mathbf{e}) &= \dot{\mathbf{e}}^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \dot{\mathbf{e}} \\ &= \mathbf{e}^T \sum_{i=1}^N h_i(\hat{\mathbf{x}}) (\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i - \mathbf{C}^T \mathbf{L}_i^T \mathbf{P} - \mathbf{P} \mathbf{L}_i \mathbf{C}) \mathbf{e} \\ &\quad + 2\Delta^T \mathbf{P} \mathbf{e} \stackrel{!}{<} 0 \end{aligned} \quad (22)$$

Die Matrix im quadratische Term in (22) wird im folgenden durch eine negativ definite Matrix ersetzt, was zur Reduktion der später zu lösenden LMI's führt.

$$\mathbf{Q} \geq \mathbf{0} \quad (23)$$

$$\mathbf{e}^T \sum_{i=1}^N h_i(\hat{\mathbf{x}}) (\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i - \mathbf{C}^T \mathbf{L}_i^T \mathbf{P} - \mathbf{P} \mathbf{L}_i \mathbf{C}) \mathbf{e} \leq -\mathbf{e}^T \mathbf{Q} \mathbf{e} \quad (24)$$

Da $\sum_{i=1}^N h_i(\hat{\mathbf{x}}) = 1$ gilt $\mathbf{Q} = \sum_{i=1}^N h_i(\hat{\mathbf{x}}) \mathbf{Q}$ und (25) ist erfüllt falls

$$\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i - \mathbf{C}^T \mathbf{L}_i^T \mathbf{P} - \mathbf{P} \mathbf{L}_i \mathbf{C} \leq -\mathbf{Q}, \forall i. \quad (25)$$

Der zweite, nicht-quadratische Term in (22) wird ersetzt durch

$$2\Delta^T \mathbf{P} \mathbf{e} \leq \alpha^2 \|\mathbf{e}\|_2^2 + \mathbf{e}^T \mathbf{P} \mathbf{P} \mathbf{e}. \quad (26)$$

Hier wurde die Annahme (20) benutzt, sowie die Ungleichung

$$2\mathbf{a}^T \mathbf{b} \leq \|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2. \quad (27)$$

Für die Herleitung dieser Ungleichung kann man das Skalarprodukt sowie die quadrierten Normen als Summe ausschreiben, und die Binomialformel $(a_i - b_i)^2 \geq 0$ einsetzen.

Setzt man (25) und (26) in (22) ein, so erhält man

$$\dot{V} \leq \mathbf{e}^T (-\mathbf{Q} + \alpha^2 \mathbf{I} + \mathbf{P} \mathbf{P}) \mathbf{e} \stackrel{!}{<} 0. \quad (28)$$

Um die Bedingungen (23), (25) und (28) als lineare Matrixungleichung zu formulieren, werden die bilinearen Terme in (25) zusammengefasst zu $\mathbf{M}_i = \mathbf{P} \mathbf{L}_i$, und der bilineare Term $\mathbf{P} \mathbf{P}$ in (28) wird durch das Schur-Komplement aufgelöst. Zusätzlich wird α^2 durch λ ersetzt und maximiert, um eine möglichst hohe Lipschitz-Konstante zu erhalten. Das zu lösende semidefinite Programm ergibt sich somit zu:

Maximiere λ unter den Nebenbedingungen:

$$\lambda > 0 \quad (29)$$

$$\mathbf{P} > \mathbf{0} \quad (30)$$

$$\mathbf{Q} \geq \mathbf{0} \quad (31)$$

$$\begin{bmatrix} \mathbf{Q} - \lambda \mathbf{I} & \mathbf{P} \\ \mathbf{P} & \mathbf{I} \end{bmatrix} > \mathbf{0} \quad (32)$$

$$\mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i - \mathbf{C}^T \mathbf{M}_i^T - \mathbf{M}_i \mathbf{C} \leq -\mathbf{Q}, \forall i. \quad (33)$$

Die gesuchten Matrizen \mathbf{L}_i lassen sich nach dem Lösen der LMI's berechnen durch $\mathbf{L}_i = \mathbf{P}^{-1} \mathbf{M}_i$.

4.1 Sliding-Mode-Beobachter

Der in diesem Abschnitt vorgestellte Beobachter ist in [3] und [4] vorgestellt und ist eine Erweiterung des bereits vorgestellten Thau-Luenberger-Beobachters um einem schaltenden Eingang, mit dem Ziel einen Teil der Modellungenauigkeiten zu korrigieren. Das Modell der Strecke wird in diesem Fall um Ungenauigkeiten $\boldsymbol{\xi}$ erweitert, sodass das betrachtete System nun

$$\dot{\mathbf{x}} = \sum_{i=1}^N h_i(\mathbf{x}) (\mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u} + \mathbf{d}_i) + \mathbf{D} \boldsymbol{\xi} \quad (34)$$

lautet. Die zugehörige Beobachterdynamik wird als

$$\dot{\hat{\mathbf{x}}} = \sum_{i=1}^N h_i(\hat{\mathbf{x}}) (\mathbf{A}_i \hat{\mathbf{x}} + \mathbf{B}_i \mathbf{u} + \mathbf{d}_i + \mathbf{L}_i (\mathbf{y} - \mathbf{C} \hat{\mathbf{x}})) + \mathbf{D} \boldsymbol{\nu} \quad (35)$$

angesetzt. Es können allerdings nur Ungenauigkeiten behoben werden, welche direkt am Ausgang sichtbar sind. Dies bedeutet, es muss eine Koordinatentransformation $\mathbf{z} = \mathbf{T} \mathbf{x}$ geben so, dass

$$\mathbf{T} \mathbf{D} = \begin{bmatrix} \mathbf{0} \\ \mathbf{D}_2 \end{bmatrix} \quad (36)$$

$$\mathbf{C} \mathbf{T}^{-1} = [\mathbf{0} \quad \mathbf{I}]. \quad (37)$$

Wie beim Thau-Luenberger-Beobachter muss die Annahme (20) an die Lipschitz-Stetigkeit getroffen werden. Zusätzlich muss eine Annahme für die obere Grenze der Norm der Ungenauigkeit $\boldsymbol{\xi}$ festgelegt werden

$$\eta \leq \|\boldsymbol{\xi}\| \quad (38)$$

Es sei

$$\boldsymbol{\Xi} = \begin{bmatrix} \mathbf{N}^T \\ \mathbf{D}_2 \\ \mathbf{0} \end{bmatrix} \quad (39)$$

mit $N_{\mathbf{D}_2}$ Basis von $\text{Kern}(\mathbf{D}_2)$. Es lässt sich zeigen, dass durch die Wahl von

$$\boldsymbol{\nu} = \begin{cases} \eta \frac{\mathbf{D}_2^T \mathbf{P}_2 \mathbf{e}_y}{\|\mathbf{D}_2^T \mathbf{P}_2 \mathbf{e}_y\|} & , \text{ falls } \mathbf{D}_2^T \mathbf{P}_2 \mathbf{e}_y \neq \mathbf{0} \\ \mathbf{0} & , \text{ sonst} \end{cases} \quad (40)$$

und der Lösung der LMI's

maximiere λ unter den Nebenbedingungen:

$$\begin{aligned} \lambda &> 0 \\ \mathbf{P} &> \mathbf{0} \\ \mathbf{Q} &\geq \mathbf{0} \\ \mathbf{A}_i^T \mathbf{P} + \mathbf{P} \mathbf{A}_i - \mathbf{C}^T \mathbf{M}_i^T - \mathbf{M}_i \mathbf{C} &\leq -\mathbf{Q}, \forall i \\ \begin{bmatrix} \mathbf{Q} - \lambda \mathbf{I} & \mathbf{P} \\ \mathbf{P} & \mathbf{I} \end{bmatrix} &> \mathbf{0} \end{aligned} \quad (41)$$

mit

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_3 \boldsymbol{\Xi} \\ \boldsymbol{\Xi}^T \mathbf{P}_3^T & \mathbf{P}_2 \end{bmatrix}, \quad \mathbf{M}_i = \mathbf{P} \mathbf{L}_i, \quad \lambda = \alpha^2 \quad (42)$$

der Beobachterfehler asymptotisch stabil im Ursprung ist. Die zu lösenden LMI's sind in [4] hergeleitet, wobei die Herleitung wieder auf einer quadratischen Lyapunov-Funktion basiert. Die Herleitung wird an dieser Stelle aus Platzgründen nicht wiederholt.

Die gesuchten Matrizen \mathbf{L}_i können wieder über $\mathbf{L}_i = \mathbf{P}^{-1} \mathbf{M}_i$ ermittelt werden. Da \mathbf{D} und $\boldsymbol{\xi}$ der realen Strecke unbekannt sind, wurden \mathbf{D}_2 und $\boldsymbol{\nu}$ in einer für das System passenden Größenordnung gewählt als $\mathbf{D}_2 = \mathbf{I}$ und $\eta = 4$.

5 Beobachter-Vergleich am Viertanksystem

Aus der Kombination der vorangegangenen Modellierungsansätzen und Beobachterentwürfen ergeben sich vier Beobachter, welche im Folgenden sowohl in der Simulation als auch am realen Viertanksystem verglichen werden. Die hier verwendeten Abkürzungen der verschiedenen Beobachter sind in Tabelle 2 aufgelistet. Zusätzlich wird ein Extended-Kalman-Filter zum Vergleich hinzugezogen, welcher als obs_{EKF} abgekürzt wird.

Tabelle 2: Abkürzungen für die verwendeten Beobachter

	Thau-Luenb.-Beob.	Sliding-Mode-Beob.
exakte Umformung	$obs_{TL, \text{ exakt}}$	$obs_{SM, \text{ exakt}}$
approx. TS-Darstellung	$obs_{TL, \text{ approx}}$	$obs_{SM, \text{ approx}}$

Tabelle 3: Erreichte Lipschitz-Konstante für die verschiedenen Beobachter

	$obs_{TL, \text{ exakt}}$	$obs_{TL, \text{ approx}}$	$obs_{SM, \text{ exakt}}$	$obs_{SM, \text{ approx}}$
α	0.114	0.0678	0.0800	0.0469

Der Vergleich beinhaltet

1. die maximale Lipschitz-Konstante α aus dem Optimierungsproblem und Einhaltung der Annahme (20)
2. das Abklingen eines Anfangsfehlers
3. den Beobachterfehler unter Messrauschen und
4. den Beobachterfehler am realen System durch Modell- und Messfehler.

Als Fehlermaß wird der mittlere quadratische Fehler

$$E = \sqrt{\frac{1}{N_t} \sum_{k=1}^{N_t} (x_3[kT] - \hat{x}_3[kT])^2 + (x_4[kT] - \hat{x}_4[kT])^2} \quad (43)$$

bezüglich der geschätzten Zustandsgrößen herangezogen. T beschreibt hierbei die Abtastzeit von 0.01s, und N_t die Anzahl an abgetasteten Zeitpunkten für die betrachtete Trajektorie. Der freie Parameter η des Sliding-Mode-Beobachters wurde auf 4 geschätzt.

Für die Vergleiche werden jeweils vier verschiedene Trajektorien simuliert oder am realen System abgefahren. Alle LMI's konnten in JuMP[5] mittels Mosek[1] unterhalb einer Sekunde gelöst werden.

5.1 Einhalten der Lipschitz-Annahme

Die für die jeweiligen Beobachter erreichten, maximierten Lipschitz-Konstanten sind in Tabelle 3 aufgeführt.

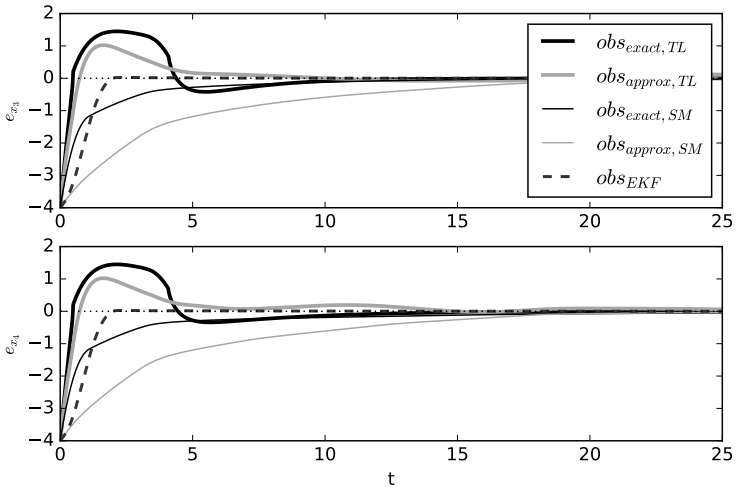


Bild 4: Fehlerabklingrate für verschiedene Beobachter.

In dem vorgestellten Beispiel wird für TS-Beobachter, basierend auf einer exakten Umformung eine um fast 70% höhere Lipschitz-Konstante erreicht.

Beim Auswerten der tatsächlich benötigten Lipschitz-Konstante zeigt sich jedoch für mehrere Trajektorien sowohl in der Simulation als auch später am realen System, dass die Annahme für alle Beobachter verletzt wird. Die Größe der tatsächlich benötigten Lipschitz-Konstanten ist dabei abhängig von der gewählten Trajektorie. Somit ist der Konvergenzbeweis basierend auf der quadratischen Lyapunov-Funktion verletzt. Trotzdem konvergieren die Beobachterfehler in den durchgeführten Untersuchungen.

5.2 Ergebnisse bei Anfangsfehler und Messrauschen

In der Simulation des Viertanksystems klingt der Anfangsfehler schneller ab für Beobachter basierend auf einem TS-Modell, welches über den Sektor-Nichtlinearitäten-Ansatz erstellt wurde. Auch zeigt der Thau-Luenberger-Beobachter eine schnellere Konvergenz als der Sliding-Mode-Beobachter. Der Beobachterfehler in einer Simulation ohne Messrauschen ist für eine Beispiel-Trajektorie in Bild 4 dargestellt, und verhält sich ähnlich für andere Trajektorien oder Anfangsfehler.

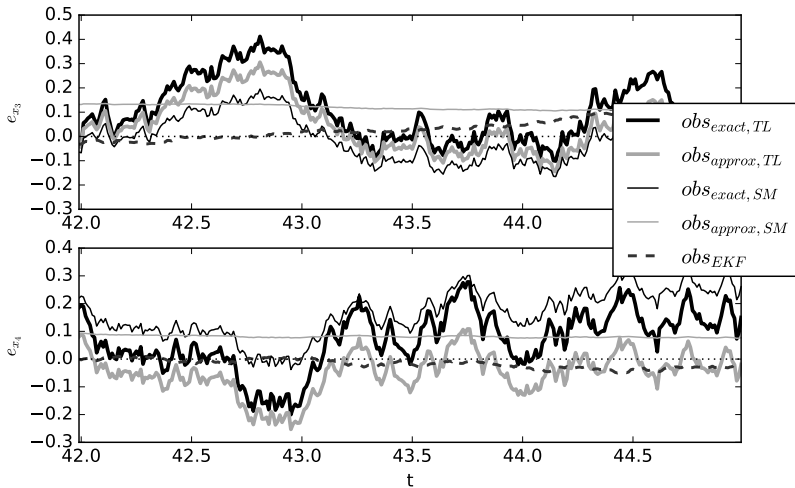


Bild 5: Ausschnitt des Beobachterfehlers bei Messrauschen.

Betrachtet man das Verhalten der verschiedenen Beobachter bei Messrauschen, so verhalten sich in der Simulation des Viertanksystems die Beobachter robuster, welche auf einer approximativen TS-Darstellung basieren. Ein Ausschnitt des Beobachterfehlers bei aufgetragenem Messrauschen $\mathbf{y}_{\text{Mess}}[kT] = \mathbf{y}[kT] + \mathcal{N}(\mathbf{0}, 3^2\mathbf{I})$ ist in Bild 5 dargestellt. Auffällig robust gegenüber Messrauschen ist hier der Sliding-Mode-Beobachter, basierend auf einem approximativen TS-Modell. Bild 6 zeigt den mittleren quadratischen Fehler über eine Beispiel-Trajektorie für unterschiedlich starkes Messrauschen. Auch hier sind die Ergebnisse ähnlich für unterschiedliche Trajektorien und der Sliding-Mode-Beobachter, basierend auf einer approximativen TS-Darstellung, zeigt eine höhere Robustheit gegenüber Messrauschen als die anderen Beobachter auf TS-Basis.

Die Parameter des Extended-Kalman-Filter wurden in den vorangegangenen Simulationen an das Messrauschen angepasst, sodass dieser einen deutlich kleineren Beobachterfehler liefert als die Beobachter auf TS-Basis, wie in Bild 6 zu erkennen.

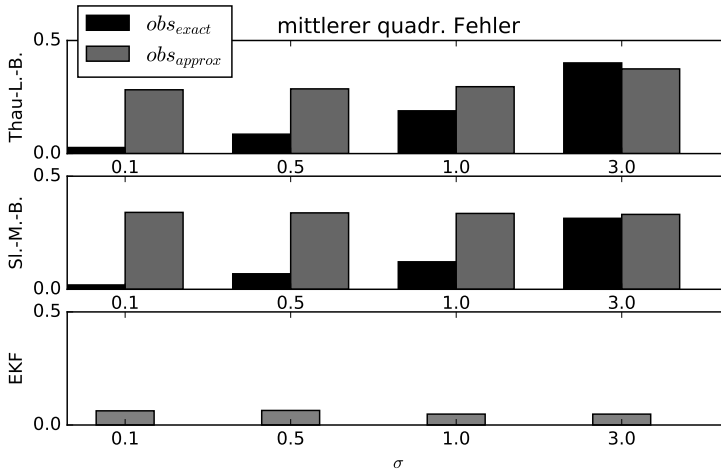


Bild 6: Mittl. quadr. Beobachterfehler über eine Beispiel-Trajektorie für unterschiedlich starkes Messrauschen und unterschiedliche Beobachter in der Simulation.

5.3 Beobachterergebnisse am realen System

Der Beobachterfehler über eine Beispiel-Trajektorie an der realen Strecke ist in Bild 7 gezeigt. Tabelle 4 zeigt den mittleren quadratischen Beobachterfehler über die vier betrachteten Trajektorien. Für das betrachtete System ist der Einfluss der unterschiedlichen TS-Modellierungen weniger ausschlaggebend als im vorherigen Abschnitt in der Simulation. In Tabelle 4 ist zu erkennen, dass die Sliding-Mode-Beobachter am realen Viertanksystem einen kleineren Beobachterfehler aufgrund des schaltenden Anteils besitzt als die Thau-Luenberger-Beobachter.

Tabelle 4: Mittlerer quad. Beobachterfehler an der realen Strecke für verschiedene Trajektorien, in cm.

Beobachter	Traj.1	Traj.2	Traj.3	Traj.4
$obs_{TL, \text{ exakt}}$	1.356	1.213	1.293	1.877
$obs_{TL, \text{ approx}}$	1.322	1.187	1.201	2.131
$obs_{SM, \text{ exakt}}$	0.831	0.936	0.836	1.700
$obs_{SM, \text{ approx}}$	0.817	0.884	0.873	1.533

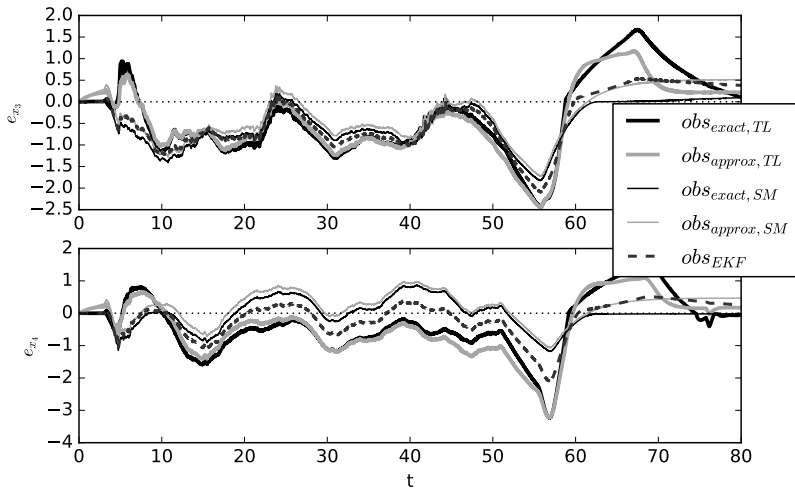


Bild 7: Beobachterfehler an einer Trajektorie an der realen Strecke.

5.4 Diskussion der Ergebnisse

Beim Vergleich der verschiedenen Ergebnisse am Viertanksystem zeigen sich unterschiedliche Vor- und Nachteile. Obwohl die Thau-Luenberger-Beobachter eine höhere Lipschitz-Konstante gegenüber den Sliding-Mode-Beobachtern erzielen, zeigt sich in den Versuchen der Sliding-Mode-Beobachter robuster gegenüber Messrauschen und Modellfehlern. Bei der Wahl der TS-Modellierung überzeugt in diesem Fall die exakte Umformung mit weniger Teilsystemen und schnellerer Fehler-Abklingrate, weist jedoch Nachteile im Bezug auf Robustheit bei Störungen und Messrauschen auf. Es soll nicht unerwähnt bleiben, dass für manche Trajektorien der Thau-Luenberger-Beobachter, insbesondere basierend auf einer exakten Umformung, deutliche Überschwinger im Beobachterfehler bei gegebenem Anfangsfehler aufzeigte, welche beim Sliding-Mode-Beobachter nicht auftraten.

6 Zusammenfassung

Im Beitrag wurden vier Beobachter entworfen und am Modell des Viertank-systems sowohl in der Simulation als auch an der realen Strecke verglichen. Es zeigt sich, dass nicht nur die Wahl des Beobachterentwurfs, sondern auch die Methode zur Überführung der Systemdynamik in ein TS-Modell starke Auswirkungen auf die Beobachterdynamik hat. Für die Anwendung am realen System waren die Unterschiede der Beobachter aufgrund der TS-Modellierung jedoch weniger ausschlaggebend als in der Simulation. An der realen Strecke spielte die Wahl des Beobachter-Entwurfsverfahrens eine bedeutendere Rolle spielte.

In dem hier vorgestellten Fall wurde ein Sliding-Mode-Beobachter, basierend auf einem exakten TS-Modell, als Favorit für den weiteren Einsatz mit einer Regelung auserwählt, aufgrund seines geringen Beobachterfehlers und geringeren Rechenanforderungen im Vergleich mit den anderen vorgestellten Beobachtern. Die Ergebnisse in diesem Beitrag sind jedoch rein experimentell, sodass eine Überführung auf andere nichtlineare Modelle in der Regel nicht ohne weiteres gültig ist. Für allgemeinere Aussagen könnte der Vergleich an weiteren Strecken durchgeführt werden. Theoretische Ergebnisse, wie sich die Wahl der Teilsysteme auf die Beobachterdynamik auswirkt gibt es bisher nicht.

Zusätzlichen könnte ein Beobachterentwurf, welcher auf einer nicht-quadratischen Lyapunov-Funktion basiert [8], zum Vergleich hinzugefügt werden.

Literatur

- [1] MOSEK ApS. „The MOSEK C optimizer API manual Version 7.0 (Revision 141)“. 2015.
- [2] P. Bergsten und R.Palm. „Thau-Luenberger observers for TS fuzzy systems“. In: *Ninth IEEE International Conference on Fuzzy Systems..* 2000.
- [3] P. Bergsten, R. Palm und D. Driankov. „Fuzzy observers“. In: *10th IEEE International Conference on Fuzzy Systems., Institute of Electrical & Electronics Engineers (IEEE)*. 2001.

- [4] P. Bergsten, R. Palm und D. Driankov. „Observers for Takagi-Sugeno fuzzy systems“. In: *IEEE Trans. Syst., Man, Cybern. B, Institute of Electrical & Electronics Engineers* . 2002.
- [5] I. Dunning, J. Huchette und M.Lubin „JuMP: A modeling language for mathematical optimization“. In: *arXiv:1508.01982 [math.OC]* . 2015.
- [6] K. Johansson. „The quadruple-tank process: a multivariable laboratory process with an adjustable zero“. In: *IEEE Transactions on Control Systems Technology, Institute of Electrical and Electronics Engineers* . 2000.
- [7] K. Tanaka und H.O. Wang. „Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach“. *Wiley-Interscience* . 2001.
- [8] D. Ichalal, B. Marx, J. Ragot und D.Maquin „Advances in observer design for Takagi-Sugeno systems with unmeasurable premise variables“. In: *20th Mediterranean Conference on Control & Automation (MED), Institute of Electrical & Electronics Engineers (IEEE)* . 2012.

Takagi-Sugeno Fuzzy Systeme für örtlich verteilte Raumklimagrößen

Tarek Aissa, Alessio Cavaterra, Steven Lambeck

Hochschule Fulda

Leipziger Str. 123, 36037 Fulda

E-Mail: {tarek.aissa,alessio.cavaterra,steven.lambeck}@et.hs-fulda.de

Kurzfassung

Die Regelung des Raumklimas ist eine wichtige Aufgabe der präventiven Konservierung. Hierbei muss meist auf dezentrale Regelungsansätze mit mobilen Endgeräten, wie Be- und Entfeuchtungsgeräte, zurückgegriffen werden. Stand der Technik bei der Modellbildung und beim Reglerentwurf ist die Betrachtung des Raumklimas als konzentriert-parametrisches System. Sowohl Messungen als auch Erfahrungswerte zeigen allerdings, dass diese vereinfachende Betrachtungsweise häufig zu unerwünschten Ergebnissen führt. Aus diesem Grund wird das Raumklimaverhalten in diesem Beitrag als verteilt-parametrisches System betrachtet. Es werden Ansätze zur Modellbildung, Systemidentifikation und Regelung vorgestellt.

1 Einführung

Das Aufgabenfeld der präventiven Konservierung befasst sich mit dem Schutz von Kulturgütern durch vorbeugende Maßnahmen [1, 2]. Um der Schädigung von Kulturgütern vorzubeugen ist eine den Materialien angepasste Lagerung von entscheidender Bedeutung. Daher ist die Gestaltung des Mikroklimas eine sehr wichtige Aufgabe der präventiven Konservierung [3]. Mit dem Begriff „*Klima*“ werden dabei sämtliche Umweltfaktoren bezeichnet, die den Zustand der Materialien merklich beeinflussen [4]. Hierzu zählen beispielsweise Schall, Licht, Temperatur oder Luftfeuchtigkeit. Aus Sicht der präventiven Konservierung wird der Temperatur und der relativen Luftfeuchte die größte Bedeutung beigemessen, wobei die relative Luftfeuchte als die entscheidende Größe angesehen wird [5].

Um das Raumklima in gewünschter Art und Weise beeinflussen zu können, stehen aktive und passive Klimatisierungsmethoden zur Verfügung [4]. Bei den passiven Methoden wird das Raumklima nur durch natürlichen Luftwechsel beeinflusst. Dabei wird versucht, die Bausubstanz so an die Umgebung anzupassen, dass das autonome Verhalten des Raumklimas den Anforderungen genügt [4]. Solche Maßnahmen werden bei Neubauten berücksichtigt, sind allerdings in historischen Gebäuden ohne größere Umbaumaßnahmen nicht zu realisieren. Deshalb wird in den meisten Fällen auf aktive Methoden zurückgegriffen, bei denen zusätzliche technische Einrichtungen zur Klimatisierung verwendet werden. Neben großen und zentralen raumlufttechnischen Anlagen stehen hierfür mobile Endgeräte, wie Be- und Entfeuchtungsgeräte zur Verfügung. Die mobilen Endgeräte haben den Vorteil, dass keine größeren Installationen, welche wiederum einen Eingriff in die Raumschubstanz darstellen, von Nöten sind und somit auch die Installationskosten deutlich geringer sind. Beide Punkte sind auf dem Gebiet der präventiven Konservierung in historischen Gebäuden wichtige Voraussetzungen [5].

Stand der Technik bei der Verwendung mobiler Endgeräte für die Feuchte-regulierung ist die konzentriert-parametrische Betrachtung des Raumklimas [6, 7]. Jedes Gerät regelt dabei auf Basis einer einzelnen Messung im Raum, was bei einer ausgeprägteren Verteilung der Klimagrößen zwangsläufig zu unzufriedenstellenden Regelungsergebnissen führt. Abbildung 1 zeigt eine Langzeitmessung in einem kulturell genutzten Raum eines Barockschlosses. In der Abbildung 1 sind die Probleme aus konservatorischer Sicht deutlich erkennbar. Zunächst zeigt die obere Abbildung die Mittelwerte aller im Raum verteilter Sensoren für relative Luftfeuchte und Temperatur. Die Verteilung der Messwerte wird jeweils in Form von Boxplots dargestellt. Bei der Betrachtung der relativen Luftfeuchte wird deutlich, dass diese stets zu hohe Werte aufweist. In der Literatur sind häufig Empfehlungen (z.B. von Restauratoren) für die relative Luftfeuchte von ca. $\varphi = 50\%$ zu finden, die konstant überschritten wird [3]. Die beiden unteren Bilder zeigen in jedem Zeitschritt die maximale Abweichung aller zehn im Raum verteilten Sensoren. Es zeigt sich, dass die relative Luftfeuchte eine starke örtliche Verteilung aufweist, wohingegen die Betrachtung der Temperatur als konzentriert-parametrisch durchaus akzeptiert werden kann.

Es besteht also die Aufgabe, die relative Luftfeuchte in einen stationär akzeptablen Bereich zu überführen, wobei die örtliche Verteilung des Raumklimas zu berücksichtigen ist.

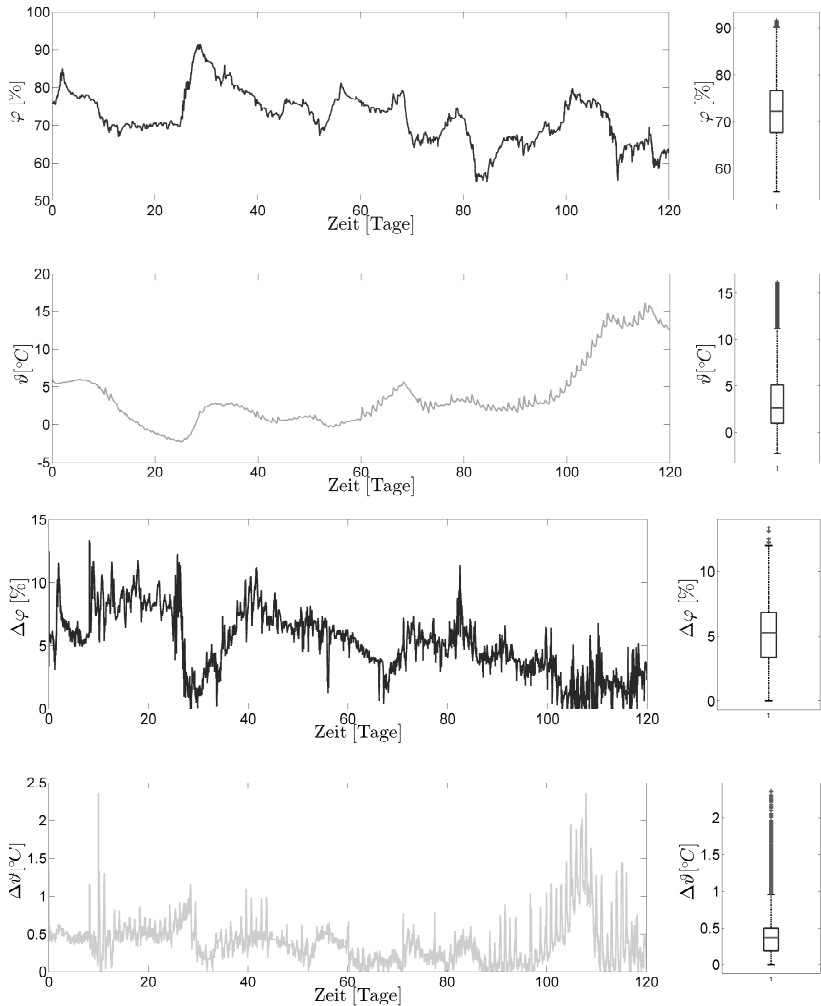


Bild 1: Oben: Mittelwerte aller Sensoren. Unten: Maximale Abweichungen der verteilten Sensoren

2 Raumklimaverhalten als verteilt-parametrisches System

Häufig werden technische Systeme als konzentriert-parametrisch betrachtet. Im Allgemeinen ist dieses Vorgehen allerdings eine Vereinfachung der realen

Zusammenhänge. Daher muss im einzelnen untersucht werden, ob eine konzentriert-parametrische Betrachtung tatsächlich ausreichend ist. Zur Beschreibung des Raumklimaverhaltens hat sich gezeigt, dass dies nicht der Fall ist [8]. Die Raumklimagrößen, insbesondere die relative Luftfeuchte, unterliegen einer ausgeprägten örtlichen Verteilung (vgl. Abbildung 1). Da die Anforderungen an das Raumklima im Bereich der präventiven Konservierung zudem sehr anspruchsvoll sind, sollte die örtliche Verteilung nicht vernachlässigt werden [8].

Ein Ansatz zur Betrachtung der örtlichen Verteilung, der bereits häufig im Rahmen der Planung von Gebäuden eingesetzt wird, sind die Computational Fluid Dynamics (CFD) Simulationen [9]. Hierbei wird das Raumklima als generelles Strömungsproblem der Fluidmechanik aufgefasst [10, 11]. Solche generellen Strömungsprobleme werden über nichtlineare partielle Differentialgleichungssysteme beschrieben. Über die Bilanzierung der Volumenströme definierter Volumen kann ein Prototyp eines solchen Strömungsproblems für beliebige Strömungsgrößen aufgestellt werden:

$$\frac{\partial}{\partial t}(\rho\phi) + \underbrace{\nabla \cdot (\rho\mathbf{u}\phi)}_{F_\phi} = \underbrace{\nabla \cdot (\Gamma\nabla\phi)}_{D_\phi} + Q_\phi \quad (1)$$

Dabei bezeichnet ϕ die Strömungsgröße. Wird hier beispielsweise das Geschwindigkeitsfeld $\phi = \mathbf{u}$ als Strömungsgröße gewählt und in Gleichung (1) eingesetzt, resultieren die bekannten Navier-Stokes Gleichungen. Dabei beschreibt der erste Term der linken Seite das dynamische Verhalten der Strömungsgröße innerhalb des Volumens. Der zweite Term beschreibt den konvektiven Fluss F_ϕ bedingt durch das Geschwindigkeitsfeld \mathbf{u} . Auf der rechten Seite beschreibt D_ϕ den diffusiven Fluss mit Diffusionskoeffizient Γ und Q_ϕ einen zusätzlichen Quellterm. Dieser Prototyp gilt zunächst erst einmal für die allgemeine Bilanzierung eines bestimmten Volumens. Für die Lösung eines solchen Strömungsproblems muss somit ein nichtlineares partielles Differentialgleichungssystem gelöst werden, was im Allgemeinen nur numerisch möglich ist. Der numerische Lösungsweg wird beispielsweise bei den zuvor genannten CFD-Simulationen besprochen. Hierbei entstehen sehr detaillierte Ergebnisse. Allerdings ist die Parametrierung solcher Simulationen schwierig und der Rechenaufwand extrem hoch. Gerade zur Reglersynthese eignen sich CFD-Simulationen somit nicht und es besteht daher die Notwendigkeit, ein reduziertes Modell zu entwickeln [7].

Durch örtliche Diskretisierungsmethoden lassen sich zunächst die partiellen Differentialgleichungen für ein definiertes Gitternetz approximieren.

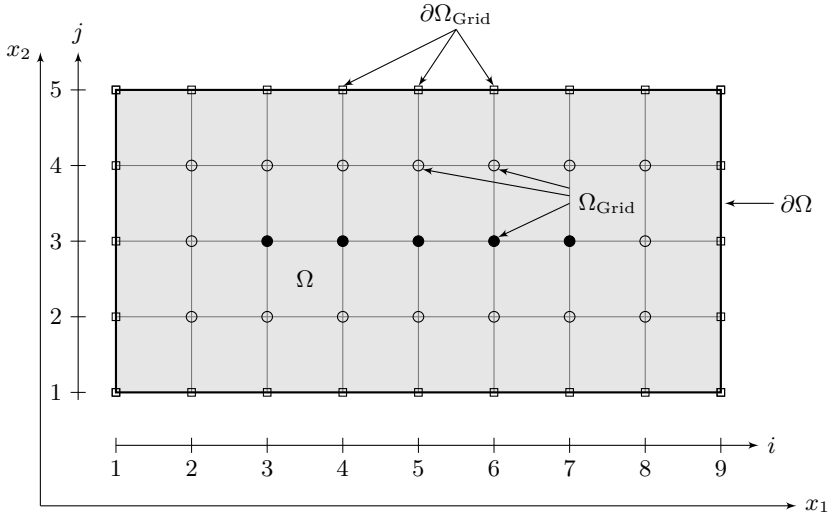


Bild 2: Festlegung der Gitterpunkte eines zweidimensionalen Problems

Bekannte Methoden hierfür sind die Finite Differenzen, Finite Elemente oder Finite Volumen [12]. Letztere werden häufig im Rahmen von CFD Simulationen verwendet, da sie komplexe geometrische Strukturen gut abbilden können. Da es sich bei dem vorliegenden Problem, der Modellierung des Raumklimas, um eher einfache geometrische Strukturen handelt, bietet sich die Verwendung der Finite Differenzen Methode an. Hierbei werden die partiellen Ableitungen nach den Ortsvariablen durch Differenzenquotienten diskretisiert:

$$\left(\frac{d\phi_i}{dx}\right) \approx \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} \quad (2a)$$

$$\left(\frac{d^2\phi_i}{dx^2}\right) \approx \frac{\phi_{i+1} + \phi_{i-1} - 2\phi_i}{(\Delta x)^2} \quad (2b)$$

Dabei ist Δx die Schrittweite des Approximationsgitters und i bezeichnet die Indizierung der Knotenpunkte. Für den örtlich zweidimensionalen Fall zeigt Abbildung 2 ein mögliches Approximationsgitter.

Dabei erfolgt die Approximation in einem definierten Arbeitsraum

$$\Omega = \{(x_1, x_2) | r_1 < x_1 < r_2, r_3 < x_2 < r_4\} \quad (3)$$

zwischen den Grenzen r_1, \dots, r_4 . Zwar ist eine äquidistante Einteilung, wie sie in Abbildung 2 gewählt wurde nicht zwingend, sie erleichtert die Handhabung allerdings erheblich. Für die Schrittweiten bei einer festgelegten Anzahl an Gitterpunkten N_{x_1} und N_{x_2} folgt:

$$\Delta x_1 = \frac{r_2 - r_1}{N_{x_1}} \quad \Delta x_2 = \frac{r_4 - r_3}{N_{x_2}} \quad (4)$$

Das Approximationsgitter ist demnach wie folgt definiert:

$$\Omega_{\text{Grid}} = \{(x_{1,i}, x_{2,j}) \in \Omega \mid x_{1,i} = r_1 + i\Delta x_1, x_{2,i} = r_3 + j\Delta x_2, i, j \in \mathbb{Z}\} \subset \Omega \quad (5)$$

Dabei wird zwischen regulären inneren Punkten (ausgefüllte Kreise), welche wiederum von inneren Punkten umgeben sind, und irregulären inneren Punkten (unausgefüllte Kreise), die Randpunkte (Rechtecke) als Nachbarn haben, unterschieden.

Mit Hilfe des Approximationsgitters und der Differenzenquotienten lassen sich die partiellen Differentialgleichungen zumindest an jedem inneren Knotenpunkt annähern. Für den Fall der irregulären Punkte müssen, wie bei verteilt-parametrischen Systemen üblich, Randbedingungen definiert werden. Betrachtet wird zunächst beispielhaft der eindimensionale Fall der Temperaturverteilung auf einer Eisenstange. Die Eisenstange liegt dabei in einer beheizbaren Flüssigkeit [13]. Das System lässt sich durch folgende nichtlineare partielle Differentialgleichung beschreiben:

$$\frac{\partial y(x, t)}{\partial t} = \frac{\partial^2 y(x, t)}{\partial x^2} + \beta_T e^{-\frac{\gamma}{1+y(x,t)}} + \beta_U (b(x)u(t) - y(x, t)) - \beta_T e^{-\gamma} \quad (6)$$

Die Temperatur am Ort x ist durch $y(x, t)$ gegeben, β_T und β_U sind Wärmeübertragungskoeffizienten. Die Wirkung der Stellgröße wird durch die Funktion $b(x) = 0, 1\beta_U \cos(0, 1x)$ beschrieben. wobei die Stellgröße auf die Temperatur der Flüssigkeit wirkt. Für den betrachteten Arbeitsraum gilt $\Omega = \{x \mid r_1 < x < r_2\} = \{x \mid 0 < x < \pi\}$. Am geometrischen Rand des Problems gilt die folgende Randbedingung:

$$y(0, t) = y(\pi, t) = 0 \quad (7)$$

Für eine beispielhafte Diskretisierung in $N_x = 7$ Intervalle folgt eine Schrittweite von:

$$\Delta x = \frac{r_2 - r_1}{N_x} = \frac{\pi - 0}{7} = \frac{\pi}{7} \quad (8)$$

Somit folgt für das Approximationsgitter:

$$\Omega_{\text{Grid}} = \{x \in \Omega \mid x = r_1 + i\Delta x, i \in \mathbb{Z}\} \subset \Omega \quad (9)$$

Die Anwendung der Differenzenquotienten auf das System aus Gleichung (6) ergibt einen Satz nichtlinearer, aber ordinärer Differentialgleichungen:

$$\frac{\partial y_i(t)}{\partial t} = \underbrace{\frac{y_{i-1}(t) - 2y_i(t) + y_{i+1}(t)}{\Delta x^2} + \beta_U(b_i u(t) - y_i(t))}_{\text{linearer Teil}} - \underbrace{\beta_T \left(e^{-\frac{\gamma}{1+y_i(t)}} - e^{-\gamma} \right)}_{\text{nichtlinearer Teil}} \quad (10)$$

Im Zusammenhang mit dem Approximationsgitter lässt sich das örtlich diskretisierte System der inneren Punkte in Matrix-Vektor Form mit dem Koeffizienten $\varsigma = -\frac{2}{\Delta x^2} - \beta_U$ wie folgt schreiben:

$$\begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \\ \dot{y}_3(t) \\ \dot{y}_4(t) \\ \dot{y}_5(t) \\ \dot{y}_6(t) \end{bmatrix} = \begin{bmatrix} \varsigma & \frac{1}{\Delta x^2} & 0 & 0 & 0 & 0 \\ \frac{1}{\Delta x^2} & \varsigma & \frac{1}{\Delta x^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\Delta x^2} & \varsigma & \frac{1}{\Delta x^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\Delta x^2} & \varsigma & \frac{1}{\Delta x^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\Delta x^2} & \varsigma & \frac{1}{\Delta x^2} \\ 0 & 0 & 0 & 0 & \frac{1}{\Delta x^2} & \varsigma \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ y_4(t) \\ y_5(t) \\ y_6(t) \end{bmatrix} + \begin{bmatrix} \beta_u b_1 \\ \beta_u b_2 \\ \beta_u b_3 \\ \beta_u b_4 \\ \beta_u b_5 \\ \beta_u b_6 \end{bmatrix} u(t) \quad (11)$$

$$+ \begin{bmatrix} -\beta_T \left(e^{-\frac{\gamma}{1+y_1(t)}} - e^{-\gamma} \right) \\ -\beta_T \left(e^{-\frac{\gamma}{1+y_2(t)}} - e^{-\gamma} \right) \\ -\beta_T \left(e^{-\frac{\gamma}{1+y_3(t)}} - e^{-\gamma} \right) \\ -\beta_T \left(e^{-\frac{\gamma}{1+y_4(t)}} - e^{-\gamma} \right) \\ -\beta_T \left(e^{-\frac{\gamma}{1+y_5(t)}} - e^{-\gamma} \right) \\ -\beta_T \left(e^{-\frac{\gamma}{1+y_6(t)}} - e^{-\gamma} \right) \end{bmatrix}$$

Das System aus Gleichung (11) ähnelt somit einem nichtlinearen Zustandsraummodell. Hierfür bietet sich die Theorie der Takagi-Sugeno (TS) Fuzzy Systeme für Modellbildung, Systemidentifikation und Reglersynthese an [14].

3 Verteil-parametrisches Takagi-Sugeno Fuzzy System

Nichtlineare dynamische Systeme lassen sich mit Hilfe der TS Fuzzy Systeme als gewichtete Kombination linearer Teilmodelle beschreiben. Der große Vorteil dieses Modelltyps ist, dass lineare Methoden der Systemidentifikation und Reglersynthese verwendet werden können. Die Vorgehensweise soll am Beispiel des verteilt-parametrischen Systems aus (6) gezeigt werden. Für den Ansatz der Sektor-Nichtlinearitäten lässt sich die nichtlineare Funktion wie folgt beschreiben [15]:

$$f_{NL}(y_i(t)) = -\beta_T \left(e^{-\frac{\gamma}{1+y_i(t)}} - e^{-\gamma} \right) = \frac{-\beta_T \left(e^{-\frac{\gamma}{1+y_i(t)}} - e^{-\gamma} \right) \cdot y_i(t)}{y_i(t)} \quad (12)$$

Folglich lässt sich auch die Matrix-Vektorform anpassen:

$$\underbrace{\begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \\ \dot{y}_3(t) \\ \dot{y}_4(t) \\ \dot{y}_5(t) \\ \dot{y}_6(t) \end{bmatrix}}_{\dot{\varrho}} = \underbrace{\begin{bmatrix} \tilde{\zeta}(y_1) & \frac{1}{\Delta x^2} & 0 & 0 & 0 & 0 \\ \frac{1}{\Delta x^2} & \tilde{\zeta}(y_2) & \frac{1}{\Delta x^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\Delta x^2} & \tilde{\zeta}(y_3) & \frac{1}{\Delta x^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\Delta x^2} & \tilde{\zeta}(y_4) & \frac{1}{\Delta x^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\Delta x^2} & \tilde{\zeta}(y_5) & \frac{1}{\Delta x^2} \\ 0 & 0 & 0 & 0 & \frac{1}{\Delta x^2} & \tilde{\zeta}(y_6) \end{bmatrix}}_{A(\varrho)} \underbrace{\begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ y_4(t) \\ y_5(t) \\ y_6(t) \end{bmatrix}}_{\varrho} + \underbrace{\begin{bmatrix} \beta_u b_1 \\ \beta_u b_2 \\ \beta_u b_3 \\ \beta_u b_4 \\ \beta_u b_5 \\ \beta_u b_6 \end{bmatrix}}_B u(t) \quad (13)$$

mit der nun nichtlinearen Koeffizientenfunktion:

$$\tilde{\zeta}(y_i) = -\frac{2}{\Delta x^2} - \beta_U + \frac{f_{NL}(y_i(t))}{y_i(t)} = -\frac{2}{\Delta x^2} - \beta_U - \frac{\beta_T \left(e^{-\frac{\gamma}{1+y_i(t)}} - e^{-\gamma} \right)}{y_i(t)} \quad (14)$$

Dabei bezeichnet ϱ den gesuchten Lösungsvektor an den Gitterpunkten. Mit Hilfe der Sektor-Nichtlinearitäten kann Gleichung (12) geschrieben werden als:

$$\begin{aligned}\Lambda(y_i(t)) &= \frac{f_{\text{NL}}(y_i(t))}{y_i(t)} = \frac{\beta_T \left(e^{-\frac{\gamma}{1+y_i(t)}} - e^{-\gamma} \right)}{y_i(t)} \\ &= \mu_1(y_i(t)) \cdot \Lambda_{\text{max}} + \mu_2(y_i(t)) \cdot \Lambda_{\text{min}}\end{aligned}\quad (15)$$

mit

$$\mu_1(z_i(t)) = \frac{\Lambda(z_i(t)) - \Lambda_{\text{min}}}{\Lambda_{\text{max}} - \Lambda_{\text{min}}} \quad \mu_2(z_i(t)) = \frac{\Lambda_{\text{max}} - \Lambda(z_i(t))}{\Lambda_{\text{max}} - \Lambda_{\text{min}}} = 1 - \mu_1(z_i(t))\quad (16)$$

Mit Hilfe der Sektorfunktionen lässt sich das System als Summe linearer Teilsysteme schreiben, indem die nichtlineare Funktion durch Λ_{min} bzw. Λ_{max} ersetzt wird.

Ebenso lässt sich das System aus Gleichung (10) approximativ durch Linearisierung der nichtlinearen Funktion in definierten Arbeitspunkten modellieren. Die Repräsentation der nichtlinearen Funktion erfolgt hierbei durch Taylor-Reihenentwicklung an festgelegten Arbeitspunkten ζ_i . Die Anzahl linearer Teilmodelle hängt somit von der Anzahl der gewählten Arbeitspunkte ab.

$$f_{\text{NL}}(y_i(t)) \approx f_{\text{NL}}(\zeta_i) + \left. \left(\frac{\partial f_{\text{NL}}}{\partial y_i} \right) \right|_{y_i=\zeta_i} \cdot (y_i(t) - \zeta_i)\quad (17)$$

Im Gegensatz zum Ansatz über Sektor-Nichtlinearitäten entsteht bei der approximativen Modellierung ein affiner Anteil, sofern nicht in einer Ruhelage linearisiert wird. Für das beispielhaft betrachtete System folgt:

$$\begin{aligned}f_{\text{NL}}(y_i(t)) &\approx \underbrace{\beta_T \left(\frac{\gamma}{(1+y_i(t))^2} \cdot e^{-\frac{\gamma}{1+y_i(t)}} \right)}_{\text{linearer Teil}} y_i(t) + \\ &\quad \underbrace{f_{\text{NL}}(\zeta_i) - \beta_T \left(\frac{\gamma}{(1+y_i(t))^2} \cdot e^{-\frac{\gamma}{1+y_i(t)}} \right)}_{\text{affiner Teil}} \zeta_i \\ &= \Lambda_{\text{lin}}^{\zeta_i} \cdot y_i(t) + \Lambda_{\text{aff}}^{\zeta_i}\end{aligned}\quad (18)$$

Auch hier lassen sich die nunmehr linearen Terme als Ersatz für die nichtlineare Funktion verwenden und es lassen sich lineare Teilmodelle bilden. Zusammenfassend gilt dann die bekannte Struktur der TS Fuzzy Systeme, wobei die Scheduling-Variablen den Werten an den Knotenpunkten $z(t) = y_i(t)$ entsprechen:

$$\dot{\varrho}(t) = \sum_{i=1}^{N_R} \eta(z(t)) [A_i \varrho(t) + \beta_i u + a_i] \quad (19)$$

Abbildung 3 zeigt die Simulation des Originalsystems, verglichen mit den Modellverläufen.

Der Ansatz der Sektor-Nichtlinearitäten liefert eine exakte Beschreibung des Systems. Modellfehler sind durch die örtliche Diskretisierung und numerische Effekte bedingt. Bei der approximativen Modellierung mit $N_R = 5$ Teilmodellen ergibt sich ein akzeptabler Approximationsfehler.

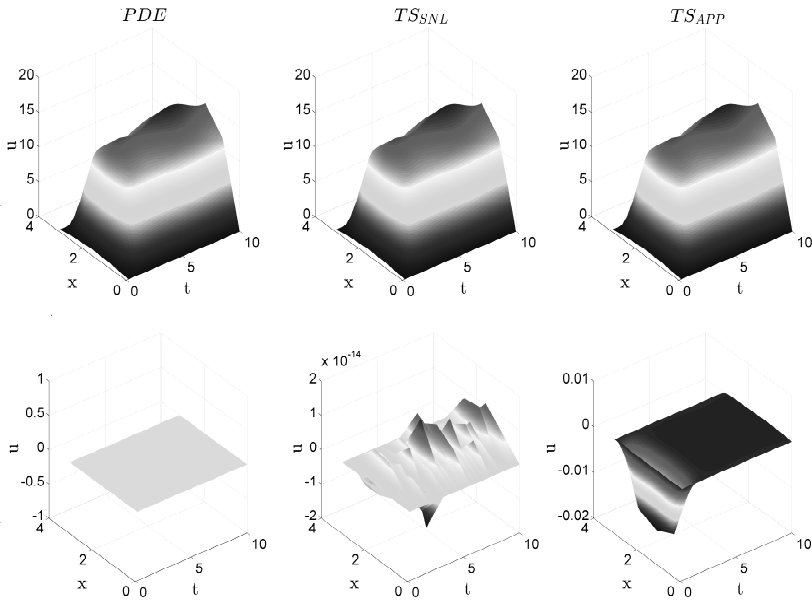


Bild 3: Oben: Originalsystem verglichen mit dem Ansatz der Sektor-Nichtlinearitäten und der approximativen Modellierung. Unten: Zugehörige Modellfehler $y_i - \hat{y}_i$

4 Modellierung und Regelung des Raumklimaverhaltens

Als grundlegenden physikalischen Zusammenhang dient der Prototyp des Strömungsproblems aus (1). Die Strömungsgröße ist im vorliegenden Fall die relative Luftfeuchte $\phi = \varphi$. Analog zu [10, 7] wird die vereinfachende Annahme getroffen, dass das unterliegende Geschwindigkeitsfeld \underline{u} als konstant angenommen werden kann. Dies ist in geschlossenen Gebäuderäumen eine angemessene Vereinfachung [16], sodass die dynamischen Gleichungen für die Strömungsgeschwindigkeiten nicht benötigt werden. Die Analyse der Raumtemperatur zeigte zudem, dass eine gemittelte Raumtemperatur $\bar{\vartheta}$ als ausreichend betrachtet werden kann (siehe Abbildung 1). Zudem wird das Außenklima φ_a in Form eines Quellterms Q mit in das Modell eingebaut. Es folgt für die relative Luftfeuchte:

$$\begin{aligned} \rho(\cdot) \frac{\partial \varphi}{\partial t} = & \Gamma(\cdot) \frac{\partial^2 \varphi}{\partial x^2} + \Gamma(\cdot) \frac{\partial^2 \varphi}{\partial y^2} + \Gamma(\cdot) \frac{\partial^2 \varphi}{\partial z^2} \\ & - \rho(\cdot) v_x \frac{\partial \varphi}{\partial x} - \rho(\cdot) v_y \frac{\partial \varphi}{\partial y} - \rho(\cdot) v_z \frac{\partial \varphi}{\partial z} + Q_\varphi(\cdot) \end{aligned} \quad (20)$$

Bei den Koeffizienten, beispielsweise der Dichte $\rho(\cdot)$ oder dem Difussionskoeffizienten $\Gamma(\cdot)$ handelt es sich um Funktionen, die von den Strömungsvariablen selbst oder der Temperatur abhängen können. Sie sind des Weiteren eine Quelle für nichtlineares Systemverhalten. Nach Anwendung der örtlichen Diskretisierungsverfahren lässt sich die relative Feuchte für einen Knotenpunkt φ_i wie folgt angeben:

$$\begin{aligned} \dot{\varphi}_{d_x, d_y, d_z} = & \alpha_{d_x, d_y, d_z} \varphi_{d_x, d_y, d_z} + \alpha_{d_x-1, d_y, d_z} \varphi_{d_x-1, d_y, d_z} + \\ & \alpha_{d_x+1, d_y, d_z} \varphi_{d_x+1, d_y, d_z} + \alpha_{d_x, d_y-1, d_z} \varphi_{d_x, d_y-1, d_z} + \\ & \alpha_{d_x, d_y+1, d_z} \varphi_{d_x, d_y+1, d_z} + \alpha_{d_x, d_y, d_z-1} \varphi_{d_x, d_y, d_z-1} + \\ & \alpha_{d_x, d_y, d_z+1} \varphi_{d_x, d_y, d_z+1} + \beta_{1, d_x, d_y, d_z} u + \beta_{2, d_x, d_y, d_z} \bar{\vartheta} + \\ & \beta_{3, d_x, d_y, d_z} \varphi_a + \aleph_{d_x, d_y, d_z}(\cdot) \end{aligned} \quad (21)$$

Die Indizierung der Gitterpunkte zeigt Abbildung 4. Die Koeffizienten α und β ergeben sich aus den linearen Anteilen der Strömungsgröße bzw. der externen Einflüsse. Das nichtlineare Verhalten wird in $\aleph(\cdot)$ zusammengefasst. Ein ähnliches Vorgehen findet sich bei der Raumklimabetrachtung durch zonale Modelle (vgl. [7]), allerdings werden die nichtlinearen Zusammenhänge dabei nicht betrachtet. Gleichung (21) lässt sich erheblich übersichtlicher in einer Matrix-Vektor Form für alle Knotenpunkte des

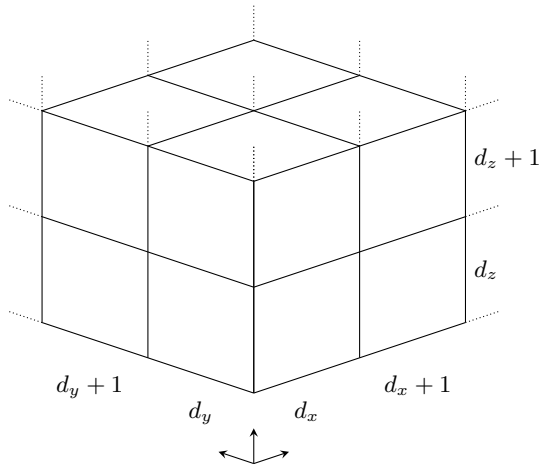


Bild 4: Indizierung der Gitterpunkte eines dreidimensionalen Gitters

Approximationsgitters aus Abbildung 4 formulieren. Hierfür wird zunächst ein Lösungsvektor ϱ eingeführt:

$$\varrho = \begin{bmatrix} \varphi_{1,1,1} \\ \varphi_{1,1,2} \\ \vdots \\ \varphi_{1,1,n_z} \\ \varphi_{1,2,1} \\ \vdots \\ \varphi_{1,n_y,n_z} \\ \vdots \\ \varphi_{n_x,n_y,n_z} \end{bmatrix} \quad (22)$$

Dementsprechend lässt sich Gleichung (21) in Matrix-Vektor Form kompakt darstellen:

$$\dot{\varrho} = A\varrho + B_c u_c + B_{uc} u_{uc} + \mathfrak{N}(\cdot) \quad (23)$$

In A finden sich die Koeffizienten der örtlichen Diskretisierung α wieder, B_c beschreibt den Einfluss der regelbaren Aktoren und B_{uc} fasst die gemittelte Raumtemperatur und das Außenklima zusammen. Die nichtlinearen Zusammenhänge beschreibt wieder $\mathfrak{N}(\cdot)$. Die Darstellung ähnelt somit einem nichtlinearen Zustandsraummodell. Im eindimensionalen Fall aus Gleichung

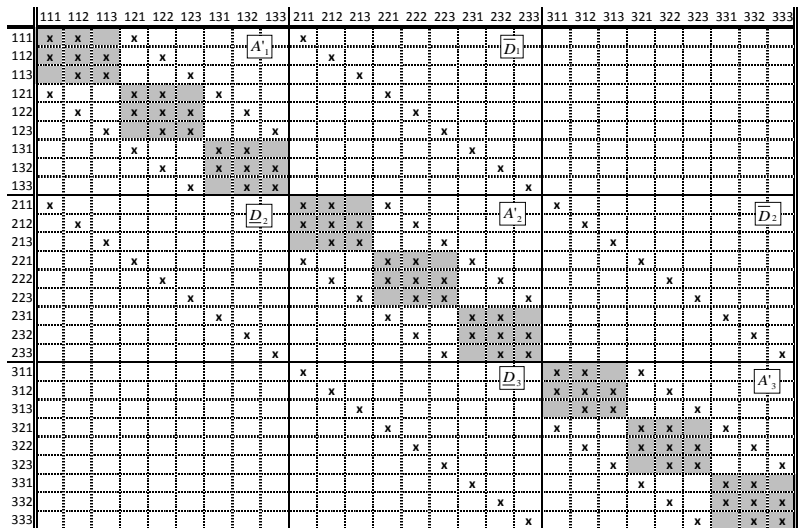


Bild 5: Besetzungsschema der blocktridiagonalen Systemmatrix A .

(13) ist die Matrix A durch eine Tridiagonal-Matrix gegeben, wie es bei der Anwendung örtlicher Diskretisierungsverfahren häufig der Fall ist. Im dreidimensionalen Fall ist die Besetzung der Matrix A weniger trivial, da sich die Zahl der örtlichen Nachbarn erhöht. Hierbei entsteht eine Blocktridiagonal Matrix (siehe Abbildung 5).

$$A = \begin{bmatrix} A'_1 & \overline{D}_1 & 0 & \cdots & 0 \\ \underline{D}_2 & A'_2 & \overline{D}_2 & \ddots & \vdots \\ 0 & \underline{D}_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \overline{D}_{n-1} \\ 0 & \cdots & 0 & \underline{D}_n & A'_n \end{bmatrix} \quad (24)$$

Das nichtlineare Modell aus Gleichung (23) lässt sich in gewohnter Form wie in (19) schreiben als TS Fuzzy System. Für den Fall der Raumklima-modellierung und Regelung ist die Bestimmung der Parameter des Modells in realen Anwendungen allerdings analytisch kaum möglich. Dafür sind die Abhängigkeiten der Parameter von den Gebäudeeigenschaften zu vielseitig. Daher muss hier auf Methoden der datengetriebenen Modellbildung zurückgegriffen werden. Ein leistungsstarkes Verfahren für TS Fuzzy Systeme ist

LOLIMOT [17, 18]. LOLIMOT ist ein Algorithmus, der iterativ eine wachsende Baumstruktur aus linearen Teilmodellen aufbaut. Der Baum wächst dabei in die Richtung der bis dato schlechtesten Teilmodellperformance. Die Modellparameter lassen sich durch eine Least-Squares Schätzung bestimmen. Da sich die globale Modellperformance somit iterativ verbessert, kann der Algorithmus solange ausgeführt werden, bis die gewünschte Modellgüte, bewertet anhand von bestimmten Gütemaßen, erreicht wurde. Die Anzahl der linearen Teilmodelle ergibt sich somit automatisch und muss nicht vorher festgelegt werden. Als Gütemaße werden häufig der mittlere quadratische Fehler (RMSE) oder sogenannte Informationskriterien, wie das Akaike Informationskriterium (AIC) oder das Bayesche Informationskriterium (BIC) [18] verwendet. Ein Vorteil des BIC ist, dass neben der Modellgüte auch die Modellkomplexität anhand der Anzahl zu optimierender Parameter bewertet wird. Die Anzahl freier Parameter im Falle der örtlich-verteilten Raumklimamodellierung ist neben der Anzahl der linearen Teilmodelle N_{TS} auch noch abhängig von der Anzahl an betrachteten Knotenpunkten N_x . Da die Anzahl der betrachteten Knotenpunkte in der realen Applikation der Anzahl an Messstellen entspricht, bleibt die Frage zu klären, wieviele Messstellen benötigt werden um das Raumklima angemessen modellieren zu können. Mit Hilfe des BIC wird hier eine Empfehlung gegeben, die einen Kompromiss aus Modellgüte und Modellkomplexität darstellt. Die Anwendung sowie die Ergebnisse werden im folgenden Kapitel dargestellt.

4.1 Ergebnisse

Als Applikation für die Regelung des Raumklimas dient das Barockschloss „Fasanerie“ in der Nähe von Fulda. Für die folgenden Untersuchungen wurde ein kulturell genutzter Raum ausgewählt und zunächst über einen Zeitraum von einem halben Jahr vermessen. Zudem wurden die Außenklimadaten erfasst. Die Ergebnisse der Messung wurden bereits weiter oben in Abbildung 1 dargestellt. Zum einen ist der Verlauf von relativer Luftfeuchte φ und Temperatur ϑ als Mittelwert aller Sensoren in den oberen beiden Abbildungen dargestellt. Es zeigt sich, dass die relative Luftfeuchte konstant über einem gewünschten Wert von $\varphi_w = 50\%$ liegt. Die unteren beiden Abbildungen zeigen die maximale Abweichung aller Sensoren zu jeder Messung, ebenfalls für relative Luftfeuchte und Temperatur, als Indikator für die örtliche Verteilung. Deutlich erkennbar sind die starken örtlichen Schwankungen von bis zu $\Delta_x \varphi = 14\%$ bei der relativen Luftfeuch-



Bild 6: Das Schloss Fasanerie bei Fulda dient als Anwendungsfall.

te. Deutlich geringer fallen die örtlichen Schwankungen bei der Temperatur aus, was wie bereits erwähnt, ein Grund dafür ist, dass die Temperatur als gemittelter Wert betrachtet wird. Um die Regelungen später vergleichen zu können, wird der Raum aus obiger Messung in einer CFD-Simulation nachgebildet. Hierfür wurde das Modul Fluent der Ansys Workbench 16.2 verwendet. Der Vorteil für die nachfolgenden Untersuchungen besteht in der Reproduzierbarkeit gleichbleibender Umgebungsbedingungen, sowie in der hohen Anzahl an auswertbaren Knotenpunkten. Diese liegt bei CFD Simulationen mit $N_x \gg 10.000$ deutlich höher als es in der praktischen Anwendung realisierbar ist. Zunächst werden die Ergebnisse bezüglich der Modellbildung dargestellt. Abbildung 7 zeigt das BIC für variable Teilmodellanzahl N_{TS} und Knotenanzahl N_x , wobei die Knoten jeweils möglichst äquidistant im Raum verteilt wurden.

Das vorliegende Gütegebirge weist ein globales Minimum bei $N_x = 15$ und $N_{TS} = 4$ auf, was als Empfehlung für die reale Anwendung dienen kann. Zudem zeigt Abbildung 8 die Verteilung der Modellfehler $e = y - \hat{y}$, sowie die Verteilung der maximalen örtlichen Abweichung.

Das Modell weist sehr gute Approximationseigenschaften mit einem Mittelwert von $\bar{e}_{y-\hat{y}} = 0,15\%$, bei einer Standardabweichung von $\sigma = 2,13\%$. Auch die örtliche Verteilung wird quantitativ korrekt wiedergegeben.

Für die datengetriebene Modellierung wurden zwei Aktoren im Raum platziert, welche durch ein amplitudenmoduliertes Pseudo Random Binary Signal (APRBS) betrieben wurden. Diese beiden Aktoren stehen im Folgenden auch zur Regelung der relativen Feuchte im Raum zur Verfügung.

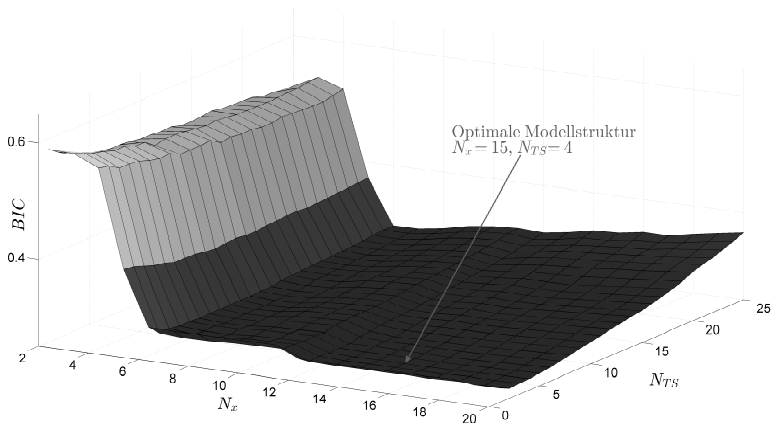


Bild 7: Auswertung des BIC für die Anzahl an Knotenpunkten N_x und die Anzahl linearer Teilmodelle N_{TS} .

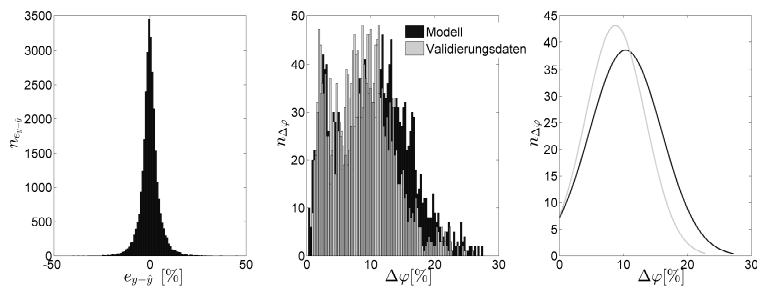


Bild 8: Verteilung der Modellfehler (links) und der maximalen örtlichen Abweichung als Histogramm (mitte) und als abgeschätzte Verteilungsfunktion (rechts).

Dabei handelt es sich um mobile Be- und Entfeuchtungsgeräte. Da sich bei zwei Aktoren und $N_x = 15$ ein nichtquadratisches, unteraktuiertes System ergibt, wird zur Regelung ein PDC-Regler mit einer Erweiterung zur Sollwertfolge verwendet, indem eine realisierbare Zielruhelage x_w, u_w berechnet wird. Das Regelgesetz lautet:

$$u = \sum_{i=1}^{N_{TS}} u_{w,i} - K_i(x - x_{w_i}) \quad (25)$$

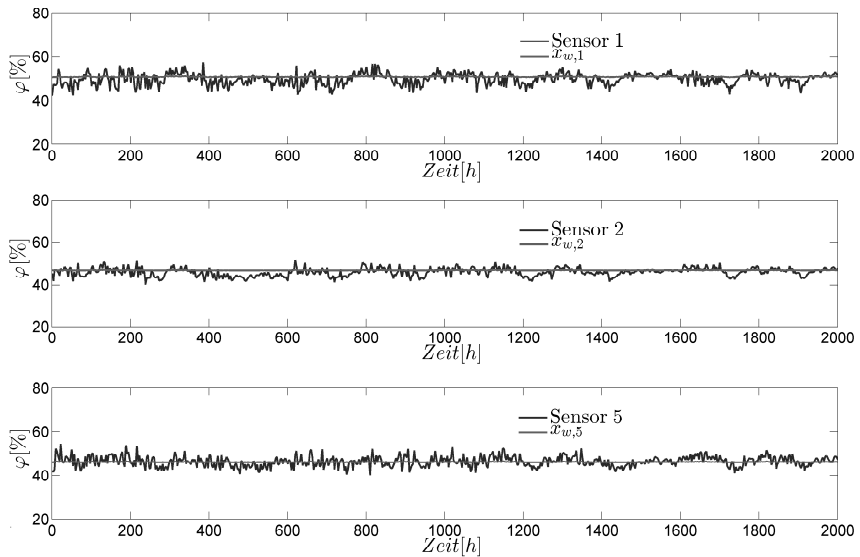


Bild 9: Auszug aus den Regelungsergebnissen.

Als Sollwert gilt für alle 15 Messstellen $w = 50\%$. Prinzipiell lässt sich die Zielruhelage eines linearen Systems wie folgt bestimmen:

$$\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \underbrace{\begin{bmatrix} A & B \\ C & 0 \end{bmatrix}}_W \begin{bmatrix} x \\ u \end{bmatrix} \quad \Leftrightarrow \quad \begin{bmatrix} 0 \\ w \end{bmatrix} = W \begin{bmatrix} x_w \\ u_w \end{bmatrix} \quad (26)$$

Im Falle nichtquadratischer Systeme ist die Matrix W allerdings nicht invertierbar und die Zielruhelage kann beispielsweise als Minimierung des quadratischen Fehlers zum Sollwert berechnet werden. Die Zielruhelage für jedes lineare Teilsystem lässt sich wie folgt berechnen (vgl. [18])

$$\begin{bmatrix} x_{w,i} \\ u_{w,i} \end{bmatrix} = \left(W_i^+ + (W_i^T W_i)^{-1} L_i^T (L_i (W_i^T W_i)^{-1} L_i^T)^{-1} L_i W_i^+ \right) \begin{bmatrix} 0 \\ w \end{bmatrix} \quad (27)$$

Wobei $L_i = [A_i \ B_i]$ und $W_i^+ = (W_i^T W_i)^{-1} W_i^T$ die Pseudoinverse beschreibt. Abbildung 9 zeigt einen Auszug aus den Regelungsergebnissen.

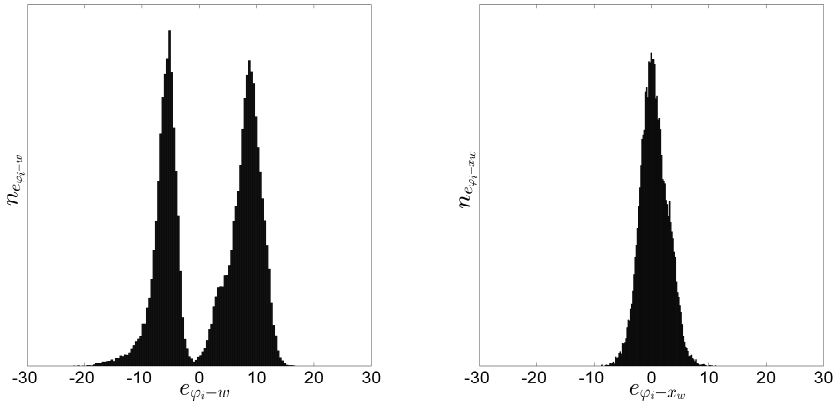


Bild 10: Verteilung der Abweichungen zum Sollwert w (links) und zur realisierbaren Ruhelage x_w (rechts).

Die rote Linie zeigt jeweils die berechnete Zielruhelage für 3 der 15 Sensoren. Für ein hochgradig gestörtes System wie das „Raumklimaverhalten“ sind die Regelungsergebnisse als gut zu bewerten. Abbildung 10 zeigt zudem die Verteilung der Regelfehler, zum tatsächlichen Sollwert w und zur realisierbaren berechneten Ruhelage x_w .

Bei der Abweichung zum tatsächlichen Sollwert ist deutlich zu erkennen, dass es sich bei der Berechnung der Zielruhelagen um eine Kompromisslösung handelt. Die Werte sind in etwa gleichverteilt über und unter dem Sollwert zu finden. In Summe weisen sie allerdings einen Mittelwert von fast null auf. Die berechneten Zielruhelagen x_w wurden hingegen gut eingeregelt.

5 Zusammenfassung und Ausblick

Im Rahmen der präventiven Konservierung sollte das Raumklima als örtlich verteiltes System betrachtet werden, um den strengen Anforderungen gerecht werden zu können. Die Eigenschaften des Raumklimas als örtlich verteiltes System lassen sich dabei durch generelle Strömungsprobleme der Fluidodynamik erfassen. Die grundlegenden physikalischen Zusammenhänge sind also durch ein System nichtlinearer partieller Differentialgleichungen gegeben. Um die aufwendige numerische Lösung, beispielsweise über

CFD-Simulationen, zu umgehen wurde in diesem Beitrag ein reduziertes Verfahren vorgestellt. Es basiert zunächst auf einer örtlichen Diskretisierung durch Finite Differenzen. Hierbei entstehen dann ordinäre nichtlineare Gleichungssysteme. Diese werden durch einen TS Fuzzy Ansatz approximiert. Da es sich hierbei um eine gewichtete Summe linearer Teilmodelle handelt, können im Anschluss Methoden der linearen Regelungstechnik angewandt werden. Für unteraktuierte und nichtquadratische Systeme, wie es bei Raumklimabetrachtungen häufig der Fall ist, wird die Sollwertfolge durch die Berechnung realisierbarer Ruhelagen realisiert. Die Ergebnisse sowohl hinsichtlich der Modellbildung als auch hinsichtlich der Regelung sind als gut zu bewerten. Da die Untersuchungen simulativ im Rahmen von CFD-Simulationen stattgefunden haben, soll im nächsten Schritt die praktische Umsetzung erfolgen.

Literatur

- [1] U. Besch. *Restauratorenentaschenbuch: Das Grundsatzpapier von Vantaa*. München, 2002.
- [2] A. Burmester. Was ist präventive konservierung? eine einföhrung. *Grundlagen der Messtechnik in der präventiven Konservierung*, pages 8–11, 2007.
- [3] A. Burmester. *Raumklima in Museen und historischen Gebäuden: Kongressband*. Fachinstitut Gebäude-Klima eV, 2000.
- [4] E. Richter, H.-M. Fischer, R. Jenisch, H. Freymuth, M. Stohrer, P. Häupl, and M. Homann. *Lehrbuch der Bauphysik: Schall-Wärme-Feuchte-Licht-Brand-Klima*. Springer-Verlag, Wiesbaden, 2008.
- [5] R. Kilian, M. Krus, and K. Sedlbauer. *Klimaanforderungen für Kunstwerke und Ausstattung historischer Gebäude*. Fraunhofer-Institut für Bauphysik, 2005.
- [6] A. Iwamae. A study on humidity distribution in a room. *Momentum*, 6(7), 2007.
- [7] X. Peng and A. Van Paassen. A state space model for predicting and controlling the temperature responses of indoor air zones. *Energy and Buildings*, 28(2):197–203, 1998.

- [8] J. Huber and K. Lerber. *Handhabung und Lagerung von mobilem Kulturgut: ein Handbuch für Museen, kirchliche Institutionen, Sammler und Archive*. Transcript Verlag, Bielefeld, 2003.
- [9] P. Jones and G. Whittle. Computational fluid dynamics for building air flow prediction - current status and capabilities. *Building and Environment*, 27(3):321–338, 1992.
- [10] S. Patankar. *Numerical heat transfer and fluid flow*. CRC Press, Boca Raton, USA, 1980.
- [11] F. White and I. Corfield. *Viscous fluid flow*. Band 3, McGraw-Hill, New York, USA, 2006.
- [12] D. Marsal. *Finite Differenzen und Elemente: Numerische Lösung von Variationsproblemen und partiellen Differentialgleichungen*. Springer-Verlag, Berlin, 2013.
- [13] P. Christofides. *Nonlinear and robust control of PDE systems: Methods and applications to transport-reaction processes*. Springer Science & Business Media, Wiesbaden, 2012.
- [14] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, (1):116–132, 1985.
- [15] H. Ohtake, K. Tanaka, and H. Wang. Fuzzy modeling via sector nonlinearity concept. *Integrated Computer-Aided Engineering*, 10(4):333–341, 2003.
- [16] X. Peng. *Modeling of indoor thermal conditions for comfort control in buildings*. PhD thesis, Technical University Delft, Niederlande, 1996.
- [17] B. Hartmann, T. Ebert, T. Fischer, J. Belz, G. Kampmann, and O. Nelles. Lmntool–toolbox zum automatischen trainieren lokaler modellnetze. *Proceedings of the 22. Workshop Computational Intelligence, Dortmund,*, pages 341–355, 2012.
- [18] O. Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer Verlag, Berlin, 2013.

Ein Nächste-Nachbarn-Ansatz zur Anomaliedetektion bei Massendaten aus kontinuierlich betriebenen Chemieanlagen

Benjamin Jäschke, Andreas Kroll

Fachgebiet Mess- und Regelungstechnik
Institute for System Analytics and Control (ISAC)
Fachbereich Maschinenbau, Universität Kassel
Mönchebergstraße 7, 34125 Kassel
E-Mail: {benjamin.jaeschke, andreas.kroll}@mrt.uni-kassel.de

Kurzfassung

Als Anomaliedetektion wird die Erkennung abnormaler Daten oder Situationen bezeichnet. Bezüglich multivariater Zeitreihen bedeutet dies, dass ungewöhnliches Verhalten der Zeitreihen erkannt werden soll. Im Gegensatz zur Ausreißerdetektion ist hier in jedem Fall ein zeitlicher Kontext zu beachten und gegebenenfalls entsprechend vorhandener Korrelationen zwischen einzelnen Zeitreihen auch ein räumlicher Kontext. Das bedeutet, dass ungewöhnliche Zustände in Form von abnormalen Signalverläufen oder in Form abnormaler Kombinationen von Zeitreihenwerten zu einem bestimmten Zeitpunkt vorliegen können. In diesem Artikel werden Sensordaten aus einer kontinuierlich betriebenen chemischen Anlage analysiert. Es handelt sich um sehr anspruchsvolle Daten, da mit Zeitreihen für etwa 1.000 Messstellen mit einer Länge von jeweils ca. 800.000 Datenpunkten ein sehr hochdimensionales Problem vorliegt. Häufig werden vergleichbare Verfahren nur auf einzelne Zeitreihen oder Trajektorien angewendet, sodass sich die Dimension zum hier verwendeten Datensatz um Größenordnungen unterscheidet. Zudem treten beim Betrieb der Anlage zahlreiche Lastwechsel auf, sodass instationäre Daten analysiert werden müssen.

In diesem Artikel wird ein Nächste-Nachbarn-Verfahren zur Erkennung abnormaler Segmente genutzt. Es handelt sich hierbei zwar um ein einfaches Verfahren, jedoch bietet es den Vorteil der Kombinierbarkeit mit beliebigen Zeitreihentransformationen und Abstandsmaßen, sodass deren Eignung untersucht werden kann, ohne dass Kollateraleffekte aufgrund eines komplexeren Ansatzes berücksichtigt werden müssen. Die Wahl geeigneter

Transformationen und Abstandsmaße ist vor dem Hintergrund der hochdimensionalen Daten essentiell. Erste Ergebnisse untermauern dies, da die Analyse einzelner Zeitreihen meist gute Ergebnisse liefert, wohingegen bei steigender Dimension neue Probleme auftreten, die durch Auffälligkeiten einzelner Signale bedingt sind oder von der abnehmenden Aussagekraft von Abständen im hochdimensionalen Raum herrühren. Eine separate Analyse einzelner Zeitreihen ist nur unter Vernachlässigung der Korrelationen zwischen den Zeitreihen möglich, was aufgrund der verfahrenstechnisch bedingten Wechselwirkungen zwischen den physikalischen Größen in einer Chemieanlage kein angemessener Ansatz ist.

Dieser Artikel behandelt die Anwendung typischer Zeitreihen-Data-Mining-Teilprozesse wie die Anwendung von Zeitreihentransformationen und Abstandsmaßen. Formbasierte Transformationen werden in Kombination mit L_p -Normen genutzt, wobei speziell auf die durch die hohe Dimensionalität der Daten entstehenden Effekte eingegangen wird. Auch die Eignung von fraktionalen Abstandsmaßen wird untersucht. Darüber hinaus wird ein multivariater Ansatz unter Verwendung der Hauptkomponentenanalyse und geeigneter Ähnlichkeitsmaße beschrieben. Untersuchungsergebnisse für reale Daten aus der zuvor angesprochenen großchemischen Anlage werden vorgestellt.

1 Einführung

Die Anomaliedetektion hat das Ziel der Erkennung von ungewöhnlichen Daten. In den meisten Anwendungen maschinellen Lernens zur Datenanalyse wird eine vergleichbare Zielstellung als Klassifikationsproblem behandelt. Es liegen in diesem Fall Instanzen normaler Daten vor und Instanzen für abnormale Daten. Letztere können einer einzelnen Klasse angehören oder aber in mehrere Klassen aufgeteilt sein, die dann verschiedene Fehlertypen beinhalten. Anhand dieser Daten kann ein Klassifikator trainiert werden, der anschließend zur Analyse neuer Daten verwendet werden kann. Ein derartiges Vorgehen beruht jedoch auf einigen Annahmen, die zur korrekten Funktion zwingend eingehalten werden müssen: Es müssen genügend Daten für alle Klassen vorhanden sein. Dies beinhaltet auch sämtliche Fehlerfälle, die erkannt werden sollen. Folglich ergibt sich direkt die zweite Annahme: Alle Fehlertypen sind bekannt. Sind diese Annahmen erfüllt, so bietet der Ansatz einer Klassifikation den Vorteil, neben der Erkennung eines Fehlers auch eine Zuordnung zu einem bestimmten Fehlertyp zu treffen.

Im hiesigen Anwendungsfall sollen Daten einer kontinuierlich betriebenen Chemieanlage überwacht werden. Der untersuchte Anlagenteil liefert Daten zu etwa 1.000 Messstellen, für die Minutenwerte aus dem Prozessinformationssystem exportiert werden. Die Daten umfassen Drücke, Durchflüsse, Temperaturen, Füllstände und chemische Analysedaten, sodass hinsichtlich der Signaldynamik physikalisch bedingt große Unterschiede zwischen einzelnen Zeitreihen bestehen. Eine detailliertere Beschreibung der Anlage findet sich in [1].

Obige Annahmen für die Verwendung eines Fehlerklassifikators sind für ein System dieser Größe nicht haltbar. Untypische Situationen treten in einer solchen Anlage meist nur ein einziges Mal auf, da anschließend Präventionsmaßnahmen ergriffen werden, die eine Wiederholung der Situation verhindern sollen. Hinreichend viele Trainingsdaten für diesen Fehlerfall sind folglich nicht vorhanden. Ferner kann bei einem komplexen technischen System auch nicht davon ausgegangen werden, dass sämtliche mögliche Fehlerfälle bekannt sind. Vielmehr ergibt sich als typische Zielstellung der Anlagenüberwachung eine Erkennung von Abweichungen vom typischen Anlagenverhalten. Im Folgenden werden solche Abweichungen als Anomalien bezeichnet, das typische Anlagenverhalten als Nominalbetrieb.

Die Motivation für den Einsatz eines Algorithmus zur Anomaliedetektion ist die Überwachung komplexer Systeme. Für diese ist der Nominalbetrieb bekannt und durch entsprechende Daten beschrieben, jedoch ist aufgrund der Komplexität keine hinreichend gute physikalische Modellbildung möglich. Die Überwachung einer solchen Anlage wird üblicherweise einem Menschen übertragen, der eine Vielzahl von Sensordaten analysieren und normale Veränderungen von abnormalen Situationen unterscheiden muss. Bei dieser Aufgabe soll ein System zur maschinellen Anomaliedetektion Unterstützung bieten, die eine ständige Überwachung sämtlicher Messstellen beinhaltet.

Ziel dieses Artikels ist die Darstellung der besonderen Herausforderungen, die Betriebsdaten komplexer technischer Systeme mit sich bringen. Im Fokus stehen dabei Wege, die hochdimensionalen Daten mit verfügbaren Methoden der Anomaliedetektion zu analysieren. Dies beinhaltet vor allem die geeignete Darstellung der hochdimensionalen Zeitreihen, sodass eine möglichst kompakte Beschreibung des Prozessverhaltens der eigentlichen Anomaliedetektion zugeführt werden kann. Dieser Schritt entspricht der Findung geeigneter Merkmale zur Beschreibung des Anlagenverhaltens.

2 Aufbau des Artikels

Die nachfolgenden Abschnitte dieses Artikels gestalten sich wie folgt: Zunächst werden grundlegende Begrifflichkeiten definiert und einige Grundlagen zum typischen Vorgehen im Bereich des Zeitreihen-Data-Mining erläutert. Anschließend wird der hier verwendete Ansatz zur Anomaliedetektion vorgestellt. Die Anwendung dieses Algorithmus wird im nachfolgenden Abschnitt beschrieben, wobei auf die Einflüsse verschiedener Transformationen und Abstandsmaße eingegangen wird. Es folgen eine Erörterung der identifizierten Schwachstelle des Vorgehens und ein Ausblick auf kommende Forschungstätigkeiten.

3 Grundlagen

3.1 Kontextdefinition

Ziel der Anomaliedetektion ist eine kontextbehaftete Beurteilung der Daten. Dies beinhaltet, dass nicht ein einzelner Datenpunkt im Vergleich zu sämtlichen Datenpunkten des Nominalbetriebs steht, sondern der Kontext des Auftretens dieses Datenpunkts berücksichtigt wird. Hier lässt sich sowohl ein zeitlicher als auch ein räumlicher Kontext definieren. Der zeitliche Kontext soll der Tatsache Rechnung tragen, dass ein und derselbe Datenpunkt sowohl normal als auch abnormal sein kann, abhängig davon, wann er auftritt und in welchem Wertebereich die Datenpunkte in der zeitlichen Umgebung liegen. Der räumliche Kontext bezieht sich auf das Verhalten anderer Zeitreihen zum selben Zeitpunkt. In komplexen Prozessen besteht in der Regel eine Vielzahl von Querbeziehungen zwischen einzelnen Signalen, sodass folglich auch Abhängigkeiten zwischen einzelnen Zeitreihen bestehen. Dies ist durch die Vielzahl von Reglern bedingt, die in einer automatisierten Anlage zum Einsatz kommen. Hierbei können untypische Kombinationen von Signalverläufen auftreten, die als Anomalie zu interpretieren sind, selbst wenn die einzelnen Zeitreihen – isoliert betrachtet – kein abnormales Verhalten aufweisen.

3.2 Abgrenzung von anderen Forschungsbereichen

In der Literatur gibt es andere Forschungsbereiche, die ähnliche Zielstellungen wie die Anomaliedetektion aufweisen und die definitiv Überschneidungen aufweisen. Aus diesem Grund soll an dieser Stelle eine vergleichende Gegenüberstellung erfolgen, bei der Gemeinsamkeiten und Unterschiede zwischen den Bereichen der Anomaliedetektion, der Ausreißerdetektion und der Erkennung neuartiger Daten („novelty detection“) herausgearbeitet werden sollen.

Die Ausreißererkenntnis bezieht sich meist auf einzelne Datenpunkte. Ein Ausreißer wird einerseits im Falle eines nicht plausiblen oder nicht physikalisch sinnhaften Wertes definiert. Dieser Fall wird häufig in der Phase der Datenvorverarbeitung abgehandelt. Die Literatur der Ausreißerdetektion beschäftigt sich jedoch auch mit Datenpunkten, die nicht zu größeren Clustern gehören oder Auffälligkeiten hinsichtlich der Punktedichte aufweisen. Als typischer Vertreter sei der Local Outlier Factor Algorithmus [2] genannt. Der Unterschied zur Anomaliedetektion besteht in der Berücksichtigung des Kontextes. Während die Ausreißerdetektion bereits einen einzigen abnormalen Datenpunkt ausfindig machen soll, wird bei der Anomaliedetektion erwartet, dass zumindest ein mehrere Datenpunkte umfassender Abschnitt unter Berücksichtigung des zeitlichen und räumlichen Kontextes untypisch ist. Diese Unterscheidung legt jedoch auch nahe, dass durch die Interpretation eines Abschnitts oder Segments als hochdimensionaler Datenpunkt ein Verfahren zur Ausreißerdetektion letztendlich eine Anomaliedetektion realisieren kann.

Die Erkennung neuartiger Daten kann eine Kontextbeachtung ähnlich der Anomaliedetektion beinhalten, setzt sich jedoch zum Ziel, jedwede Daten zu erkennen, die bisher nicht vorkamen [3]. Es werden folglich überraschende Muster gesucht, was auch in Anwendungsfeldern genutzt wird, in denen nicht zwangsläufig ein Fehler vermutet werden muss, sondern in der nur das Vorkommen neuer Situationen oder Muster von Interesse ist. Hier genügt allerdings das Vorhandensein eines hinreichend ähnlichen Musters um die Situation nicht als neuartig einzustufen, was im Falle einer Anomaliedetektion nicht erwünscht ist.

3.3 Zeitreihen-Data-Mining

Typischerweise wird an hochdimensionale Zeitreihenprobleme derart herangegangen, dass zunächst eine Vorverarbeitung und Bereinigung der Daten erfolgt. Anschließend erfolgt eine Dimensionsreduktion, was in der Zeitreihen-Literatur häufig mit dem Begriff einer „Repräsentation“ umschrieben wird. Ziel ist hier eine kompakte Beschreibung der vorliegenden Daten, sodass die Transformation einer Zeitreihe und die anschließende Analyse der sich ergebenden Repräsentation auch als Merkmalsgeneration bezeichnet werden kann. Die nun vorliegende Repräsentation kann je nach Zielstellung beispielsweise mit Methoden der Clusterbildung oder Klassifikation analysiert werden, sodass in der Regel eine Abstandsberechnung der nächste Analyseschritt ist. In diesem Abschnitt sollen einige wichtige Vertreter für Zeitreihen-Transformationen und Abstandsmaße aufgeführt werden. Diese haben maßgeblichen Einfluss auf die Leistungsfähigkeit von Anomaliedetektionsalgorithmen.

3.4 Zeitreihen-Transformationen

Zum Zweck des Zeitreihen-Data-Mining wurde in der Literatur eine Vielzahl verschiedener Transformationen vorgeschlagen. Im Rahmen dieses Artikels kann kein umfassender Überblick erfolgen, sodass an dieser Stelle auf die Übersichtsartikel von Esling und Agon [4] sowie Ding et. al. [5] verwiesen sei, die die Thematik ausführlich behandeln. In diesem Artikel soll auf formbasierte Transformationen zurückgegriffen werden, sodass das Primärziel eine Dimensionsreduktion der Zeitreihen ist, der Verlauf der Zeitreihe als solche jedoch erkennbar bleibt. Eine einfache, aber sehr leistungsfähige [6] Transformation dieser Kategorie ist die Piecewise Aggregate Approximation (PAA). Sie repräsentiert jeweils ein Segment von festzulegender Länge einer Zeitreihe durch den Mittelwert des Segments. Die Segmentlänge ist dabei anhand der Signaldynamik festzulegen. Eine variable Segmentlänge ist in diesem Zusammenhang nicht sinnvoll einsetzbar, da sie zu nicht-äquidistanten Datenpunkten führen würde, was die weitere Analyse erheblich schwieriger gestalten würde. Die PAA soll in diesem Artikel exemplarisch für eine univariate Zeitreihen-Transformation stehen, denn beim Vorliegen mehrerer korrelierter Zeitreihen wird die Transformation auf jede einzelne Zeitreihe separat angewendet. Mit diesem Vorgehen ist zwar das Ziel der Dimensionsreduktion erreichbar, jedoch erfolgt keine Berücksichtigung von Querbeziehungen zwischen einzelnen Signalen.

Aufbauend auf der PAA bietet die Symbolic Aggregate Approximation (SAX) [7] einen weiteren Transformationsschritt: Die berechneten Mittelwerte einzelner Segmente der Zeitreihe werden nur noch Intervallen zugeordnet, die wiederum mit einem Symbol repräsentiert werden. Die Anzahl der Symbole - typischerweise als Alphabetgröße bezeichnet - ist dabei ein vorab zu bestimmender Parameter. Die SAX transformiert folglich Zeitreihen zu Symbolsequenzen. Dies bietet den Vorteil, dass beispielsweise Muster einfach als bestimmte Abfolge von Symbolen definiert werden können. Ferner lassen sich die Symbole als Zustände interpretieren, was eine Vielzahl weiterer Verfahren des Data-Minings auf Zeitreihen anwendbar macht. Genannt seien hier auf Zustandsautomaten basierende Methoden, die beispielsweise von Chakraborty et. al. [8] zur Anomaliedetektion genutzt wurden. Auch die SAX wird auf univariate Zeitreihen angewendet. Eine Anwendung der SAX auf sämtliche Zeitreihen ist nicht zweckmäßig, da für n Zeitreihen mit einer Symbolanzahl a eine Zustandsanzahl von a^n folgen würde. Aus diesem Grund werden in diesem Artikel Ergebnisse unter Verwendung der PAA beschrieben.

Zu echten multivariaten Transformationen sind in der Literatur kaum Beiträge auffindbar. Eine mögliche Vorgehensweise besteht darin, unter Verwendung der Hauptkomponentenanalyse (PCA) eine möglichst kompakte Erfassung des Verhaltens sämtlicher Zeitreihen zu erzielen. Die PCA wird dabei auf gefensterte multivariate Zeitreihen angewendet. Die sich ergebende Transformationsmatrix, welche sich aus den einzelnen Hauptkomponenten zusammensetzt, wird dabei als Zeitreihen-Repräsentation der multivariaten Zeitreihe interpretiert, die in Zusammenhang mit geeigneten Abstandsmaßen demselben Algorithmus zur Anomaliedetektion zugeführt werden kann wie die formbasierte Transformation einzelner Zeitreihen. Dieser Ansatz führte jedoch in der durchgeführten Fallstudie zu keinen guten Ergebnissen, sodass in einem zweiten Schritt versucht wird, auf Basis der Transformationsmatrix nochmals eine Reduktion auf wenige Merkmale zu erreichen. Hierfür wird der Beitrag von Originalvariablen – in diesem Fall von Zeitreihen – zur ersten Hauptkomponente untersucht. Es werden die betragsmäßig größten Beiträge dazu genutzt, die entsprechende Originalvariable als einflussreich zu deklarieren. Jedes Segment der multivariaten Zeitreihe wird anschließend durch eine Liste der s einflussreichsten Messstellen repräsentiert. Die Verwendung dieser Liste als Repräsentation führt in der nachfolgend beschriebenen Fallstudie zu guten Ergebnissen.

Dennoch soll an dieser Stelle festgehalten werden, dass dieses Vorgehen nicht als konkreter Vorschlag einer multivariaten Zeitreihen-Repräsentation

gelten soll, da die PCA zur Erfassung nicht-stationärer Prozesse nicht geeignet ist. Der hier beschriebene Ansatz soll vielmehr als Beispiel dafür dienen, dass selbst mit einem sehr großen Abstraktionsgrad noch einfache Analysen möglich sind, wie in diesem Fall eine Aussage über die Auffälligkeit des entsprechenden Abschnitts. Die Entwicklung eines konkreten Verfahrens zur multivariaten Repräsentation ist noch Gegenstand der Forschung.

3.5 Abstandsmaße

Die Verwendung von Abstands- oder Ähnlichkeitsmaßen ist für den in diesem Artikel beschriebenen Ansatz zur Anomaliedetektion unumgänglich. Auch hier muss unterschieden werden zwischen Abstandsmaßen, die für formbasierte Repräsentationen geeignet sind und Abstandsmaßen, die sich auf die im PCA-basierten Fall vorliegenden Symbolsequenzen anwenden lassen. Auch in diesem Abschnitt sei für einen breiten Überblick auf die Artikel von Esling und Agon [4] sowie Ding et. al. [5] verwiesen.

Während für univariate Zeitreihen eine Vielzahl von Abstandsmaßen existiert, muss hinsichtlich der multivariaten Analyse angemerkt werden, dass letztlich die L_p -Normen die verbleibende Alternative sind. Der Abstand d zweier Zeitreihen y_1 und y_2 der Länge N ist bei den L_p -Normen definiert als:

$$d = \sqrt[p]{\sum_{i=1}^N (y_{1,i} - y_{2,i})^p} \quad \text{mit } p \geq 1 \quad (1)$$

Auf Matrixnormen wird an dieser Stelle nur bedingt eingegangen, da die Datenmatrix im hier untersuchten Fall aus Beobachtungen einzelner Messstellen besteht. Die einzige Matrixnorm, die sich sinnhaft auf derartige Daten anwenden lässt, ist die Frobeniusnorm, die der euklidischen Distanz entspricht, wenn sämtliche Zeitreihen aneinander gehängt werden. Dieses Vorgehen ermöglicht gewiss auch die Wahl anderer p , selbst im Bereich $0 < p < 1$. Dieser Bereich wird unter dem Begriff der fraktionalen Abstandsmaße zusammengefasst, die speziell im hochdimensionalen Raum häufig als Alternative zur euklidischen Distanz genannt werden [9]. Die Erweiterung anderer Abstands- und Ähnlichkeitsmaße auf hochdimensionale Daten ist oft nicht ohne Weiteres möglich oder führt aufgrund der vorliegenden Optimierungsprobleme zu erheblichem Rechenwand.

Hinsichtlich der Abstandsmaße für Symbolsequenzen gestaltet sich die Wahl deutlich einfacher, da das Dimensionsproblem in diesem Fall bereits durch

die Transformation der Zeitreihe gelöst wurde. Es wird zum Vergleich der Messstellenlisten die Levenshtein-Distanz verwendet, die der minimalen Anzahl an Änderungen der einen Symbolsequenz entspricht, sodass diese identisch zur zweiten wird.

4 k -NN-Ansatz

4.1 Auswahl des Verfahrens

In diesem Artikel wird mit einem k -NN-Ansatz ein sehr einfaches Verfahren zur Anomaliedetektion verwendet. Dies ist so zu begründen, dass das Verfahren mit sämtlichen erdenklichen Kombinationen aus Zeitreihen-Transformationen und Abstandsmaßen funktioniert. Das Verfahren an sich bleibt dabei unverändert und einfach nachzuvollziehen, die Ergebnisse mit unterschiedlichen Zeitreihen-Repräsentationen und Abstandsmaßen bleiben vergleichbar. Komplexere Verfahren zur Anomaliedetektion bergen die Gefahr, dass Kollateraleffekte entstehen, die eine korrekte Interpretation erschweren oder verhindern.

4.2 Verfahren zur Anomaliedetektion

Ausgangspunkt sei die Repräsentation einer multivariaten Zeitreihe. Die Zeitreihe wird zunächst mit einem gleitenden Fenster segmentiert, was eine einfache Form der Berücksichtigung zeitlichen Kontextes sicherstellt. Beim k -NN-Verfahren werden nun die Abstände zwischen dem zu analysierenden Segment und sämtlichen als Nominalbetrieb deklarierten Segmenten berechnet [3]. Für $k = 1$ wird direkt der Abstand zum nächsten Nachbarn, d.h. dem Segment mit dem geringsten Abstand, als „Anomaly Score“ interpretiert. Für größere k wird entsprechend der Abstand zum k -ten Nachbarn verwendet. Größere k bringen den Vorteil mit sich, dass zwar nach wie vor ein rein distanzbasiertes Verfahren vorliegt, jedoch implizit Information über die Dichte der hochdimensionalen Datenpunkte in die Bewertung einfließt. Dies ist dadurch gegeben, dass ein Abstand d_k zum k -ten Nachbarn die Zusatzinformation beinhaltet, dass $k - 1$ hochdimensionale Datenpunkte einen Abstand $d_{k-1} < d_k$ aufweisen. Somit sind Vorteile von dichte-basierten Verfahren vorhanden, die bei einer Wahl von $k = 1$ entfallen. Im Rahmen der durchgeführten Fallstudie wurde k variiert. Jedoch zeigten sich die Vorteile größerer k nur in sehr begrenztem Maße,

was darauf zurückzuführen ist, dass die Nominalbetriebsdaten mit einem gleitenden Fenster segmentiert wurden. Direkt benachbarte Segmente sind aufgrund der meist geringen Signaldynamik einander sehr ähnlich, sodass auch die sich ergebenden Anomaly Scores kaum Unterschiede aufweisen. Dieses Phänomen konnte bis $k = 100$ beobachtet werden, sodass zugunsten des Rechenaufwands für die nachfolgend beschriebenen Ergebnisse $k = 1$ gewählt wurde.

Liegt nun ein Anomaly Score für ein Segment vor, so muss im einfachsten Fall noch ein Schwellenwertvergleich erfolgen, um zu einer Aussage zu kommen, ob das Segment als normal oder als abnormal einzustufen ist. Dieser Schwellenwert kann beispielsweise derart festgelegt werden, dass ein Abschnitt im Nominalbetrieb analysiert wird. Für sämtliche in diesem Abschnitt enthaltenen Segmente werden Anomaly Scores berechnet, sodass letztlich eine Häufigkeitsverteilung erstellt werden kann, anhand derer ein geeigneter Schwellenwert festgelegt werden kann. Hieraus ergibt sich die Forderung, dass der zur Schwellenwertfestlegung genutzte Abschnitt zwei bis drei Größenordnungen länger sein muss als die analysierten Segmente, sodass eine Erstellung der Häufigkeitsverteilung sinnvoll ist. Da die Anomaly Scores keine zahlenmäßige Bedeutung aufweisen, sondern nur relativ zueinander interpretiert werden können, wird für die nachfolgende Fallstudie stets eine normierte Darstellung gewählt, bei der der Maximalwert der beobachteten Anomaly Scores den Wert 1 annimmt und ein identischer Abschnitt in den Nominalbetriebsdaten zum Wert 0 führen würde.

5 Fallstudie

Die Untersuchungen am vorliegenden Datensatz wurden jeweils nach dem im vorherigen Abschnitt beschriebenen Schema durchgeführt. Die Daten stammen aus einer kontinuierlich 24/7 betriebenen Butadienanlage [1]. Dabei treten pro Woche etwa zwei Lastwechsel auf. Von den etwa 1.000 Messstellen sind nicht alle gleichermaßen relevant und einige zudem aufgrund zu vieler fehlender Werte nicht brauchbar. Deshalb erfolgt zur weiteren Analyse eine Reduktion auf einen Datensatz von 104 Messstellen, der sich zum einen nur auf einen Anlagenteil bezieht und zum anderen hinsichtlich der Datenqualität geprüft wurde. Exemplarisch ist in Abbildung 1 eine der 104 Zeitreihen (normiert) dargestellt, die die Anomalie in Form einer Oszillation zeigt. Es ist hervorzuheben, dass sich die Anomalie nicht auf allen Zeitreihen manifestiert und dass der Wertebereich für y_{norm} auch

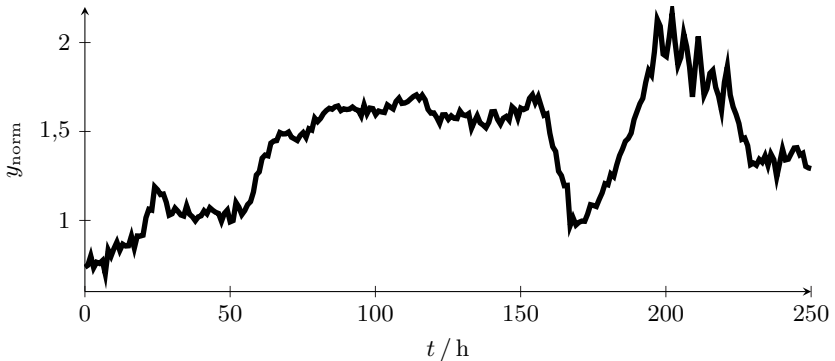


Bild 1: Beispiel-Zeitreihe aus dem verwendeten Datensatz, Lastwechsel z.B. bei $t = 60$ h, Anomalie in Form von Oszillationen bei $t = 190 \dots 230$ h

im Nominalbetrieb erreicht werden kann. Ein einfacher Schwellenwertvergleich genügt folglich nicht zur Erkennung der Anomalie. Aus Gründen der Übersichtlichkeit wurde lediglich ein kurzer Abschnitt der Zeitreihe abgebildet.

5.1 Formbasierte Anomaliedetektion

Zunächst erfolgt die Anomaliedetektion mit Hilfe der PAA und unter Anwendung der L_2 -Norm. Die PAA bewirkt in diesem Fall eine Dimensionsreduktion um den Faktor 10, sodass mit der Segmentlänge von einer Stunde ein 624-dimensionaler Datenpunkt vorliegt, der mit den Nominalbetriebsdaten verglichen wird. Abbildung 2 zeigt die berechneten Anomaly Scores, sowie den ermittelten Schwellenwert von 0,542. Dieser ergibt sich aus einem ebenfalls analysierten Nominalbetriebsabschnitt, mit dessen Hilfe eine Häufigkeitsverteilung der Anomaly Scores für normale Abschnitte erhalten wurde. Der Schwellenwert wurde derart gewählt, dass 5 % der Anomaly Scores oberhalb der Schwelle liegen, sodass eine Analyse neben dem Schwellenwertvergleich auch die Dauer der Schwellenwertüberschreitung beinhalten sollte. Die Anomalie ist folglich aufgrund des Anomaly Scores und der mehrfachen Überschreitung der Schwelle in kurzer Zeit unter Verwendung der L_2 -Norm zu erkennen.

Da trotz der Verwendung der PAA noch immer ein hochdimensionales Problem vorliegt, soll untersucht werden, ob mit Hilfe der Verwendung

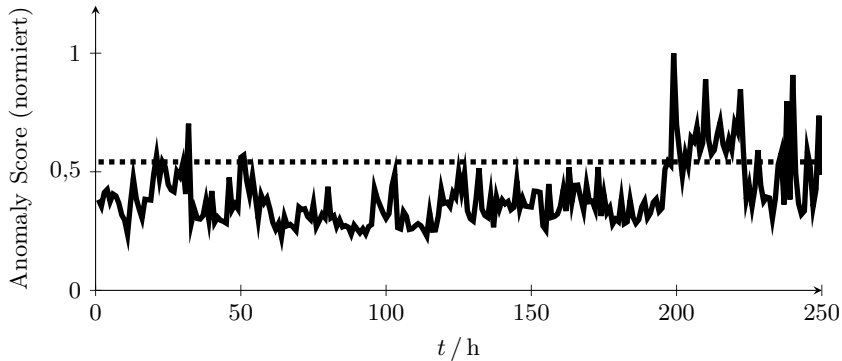


Bild 2: Ergebnisse unter Verwendung der L_2 -Norm, Anomalie bei $t = 190 \dots 230$ h

fraktionaler Abstandsmaße bessere Ergebnisse erzielt werden können. Abbildung 3 zeigt Anomaly Scores und Schwellenwert von 0,504 bei analogem Vorgehen zur L_2 -Norm unter Verwendung von $p = 0,5$. Abbildung 4 beinhaltet die Ergebnisse für $p = 0,1$ mit einem Schwellenwert von 0,472. Es wird deutlich, dass eine nennenswerte Verbesserung allein durch die Verwendung fraktionaler Abstandsmaße nicht möglich ist.

Das Dimensionsproblem lässt sich in der Phase der Abstandsberechnung mit einem solch einfachen Ansatz nicht lösen. Der Grund hierfür liegt auch in der Charakteristik der Anomalie: Sie tritt räumlich begrenzt auf, was bedeutet, dass mit zunehmender Anzahl an Messstellen, die in die Analyse einbezogen werden, der Anteil der von einer Anomalie betroffenen Zeitreihen kleiner wird. Ferner ist zu einem beliebigen Zeitpunkt mit vereinzelt Auffälligkeiten einzelner Messstellen zu rechnen, die wiederum zur Erhöhung der durchschnittlichen Anomaly Scores führen. Beide Effekte führen zu einer schlechten Abgrenzbarkeit von Anomalien und Nominalbetrieb. Jedoch stehen diese Effekte nicht direkt mit dem Konzentrationsphänomen [9] in Verbindung, sodass auch die Verwendung fraktionaler Abstandsmaße keinen adäquaten Lösungsansatz darstellt.

5.2 PCA-basierte Anomaliedetektion

Aufgrund der im vorherigen Abschnitt erwähnten Probleme sollte versucht werden, eine kompaktere Beschreibung des Prozesses zu finden, was in Form

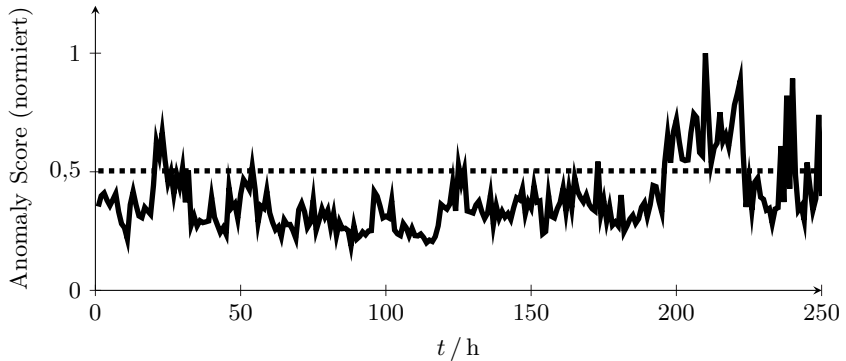


Bild 3: Ergebnisse unter Verwendung von $p = 0,5$, Anomalie bei $t = 190 \dots 230$ h

einer geeigneten Zeitreihen-Transformation geschehen kann. In diesem Abschnitt werden Ergebnisse unter Verwendung eines PCA-basierten Ansatzes vorgestellt. Dabei wurden im Gegensatz zu den formbasierten Methoden Segmente einer Länge von 10 h analysiert, da eine PCA-Berechnung für Matrizen mit mehr Spalten als Zeilen nicht möglich ist. Anschließend wird für jedes Segment eine Liste der einflussreichsten Messstellen ermittelt. Diese Symbolsequenzen werden als Repräsentation genutzt und unter Anwendung der Levenshtein-Distanz mit dem Nächste-Nachbarn-Verfahren zur Anomaliedetektion genutzt. Wichtig ist die geeignete Wahl der Listenlänge, die in den hier gezeigten Ergebnissen 15 Messstellen umfasst. Bei kürzeren Listen besteht die Gefahr, dass Lastwechsel die Listenerzeugung dominieren, sodass die Anomalie nicht erkannt wird. Zu lange Listen wiederum beinhalten Messstellen mit kleineren Auffälligkeiten, deren Einfluss durch die Verwendung der Repräsentation reduziert werden sollte.

Abbildung 5 zeigt die Ergebnisse mit PCA-basierten Listen. Der Schwellenwert wurde analog zum Vorgehen bei der formbasierten Methode berechnet und liegt normiert bei 0,8. Es zeigt sich, dass während der Anomalie offenbar andere Messstellen bestimmend für die erste Hauptkomponente sind als im Nominalbetrieb. Auch von normalen Lastwechseln ist die Anomalie unterscheidbar. Eine Untersuchung von Daten, die mehrere verschiedenartige Anomalien zeigen, ist bei dieser Repräsentation von noch größerer Bedeutung als in der formbasierten Anomaliedetektion, da das Verhalten der PCA-basierten Zeitreihen-Transformation kaum vorhersagbar ist.

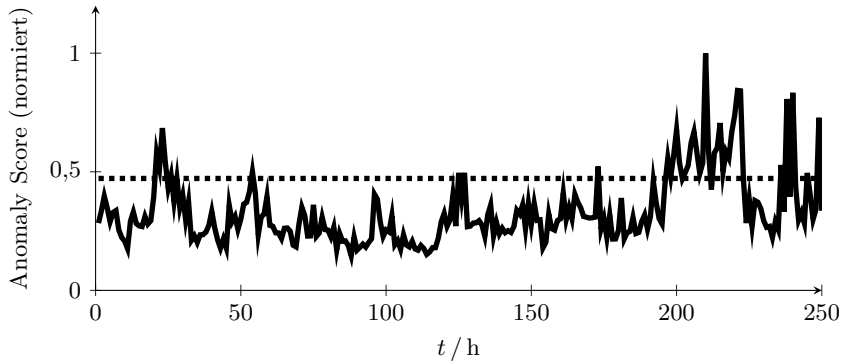


Bild 4: Ergebnisse unter Verwendung von $p = 0, 1$, Anomalie bei $t = 190 \dots 230$ h

Im hier vorliegenden Datensatz ist jedoch nur die hier gezeigte Anomalie vorhanden, sodass eine Evaluierung nicht möglich ist.

5.3 Zusammenfassung und Ausblick

Dieser Artikel stellt ein einfaches Verfahren zur Anomaliedetektion vor. Es wurde in Kombination mit verschiedenen Transformationen und Abstandsmaßen angewendet. Im Hinblick auf hochdimensionale Daten zeigt sich, dass Abstandsberechnungen ein kritischer Schritt in der Analyse sind, da für hohe Dimensionalität kaum Alternativen zu L_p -Normen verfügbar sind und das Dimensionsproblem mit speziellen Abstandsmaßen nicht adäquat gelöst werden kann. Folglich ist die Dimensionsreduktion der wichtigste Schritt in der Analyse, denn sie ermöglicht es, auch hochdimensionale Daten derart einer Abstandsberechnung zuzuführen, dass die berechneten Ergebnisse verwertbar sind. Speziell für komplexe Systeme werden die Verkopplung einzelner physikalischer Größen und somit die Querbeziehungen zwischen einzelnen Zeitreihen mit den in der Literatur der Zeitreihenanalyse üblichen Transformationen jedoch nicht berücksichtigt. Ein sehr stark abstrahierender Ansatz mit Hilfe der PCA liefert hier Ergebnisse, die zeigen, dass sich Information über den Prozess auch kompakt erfassen lässt. Dennoch sei an dieser Stelle darauf hingewiesen, dass dieser Ansatz sehr viel Information vernachlässigt und bei kleinsten Änderungen der Eingabedaten gänzlich andere Ergebnisse liefern kann. Die Ergebnisse sollen jedoch motivieren,

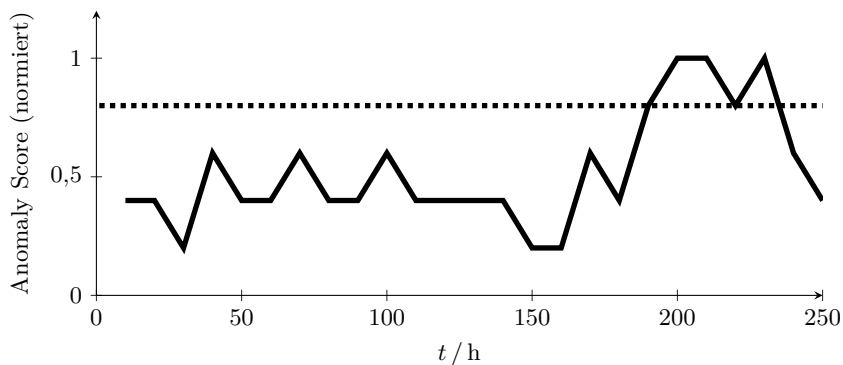


Bild 5: Ergebnisse unter Verwendung von PCA und Levenshtein-Distanz, Anomalie bei $t = 190...230$ h

dass kompakte multivariate Repräsentationen ein sinnvoller Beitrag zum Zeitreihen-Data-Mining sind.

Die weitere Forschung im Bereich der Anomaliedetektion wird sich verstärkt geeigneten Zeitreihen-Transformationen widmen. Der PCA-basierte Ansatz in diesem Artikel hat gezeigt, dass eine kompakte Erfassung des Systemverhaltens prinzipiell möglich ist. Jedoch muss ein Verfahren gefunden werden, das robuster ist als der beschriebene Ansatz und für die in großen Chemieanlagen üblicherweise anfallenden nicht-stationären Daten geeignet ist. Idealerweise liefert eine solche multivariate Zeitreihen-Repräsentation Symbolsequenzen, die als Anlagenzustände interpretierbar sind und sich beispielsweise in Kombination mit Markov-Methoden zur Anomaliedetektion eignen.

Förderung

Die Forschungstätigkeiten zu diesem Artikel erfolgten im Rahmen des BMBF-geförderten Projekts „FEE - Frühzeitige Erkennung und Entscheidungsunterstützung für kritische Situationen im Produktionsumfeld“ mit dem Förderkennzeichen 01IS14006D.

Literatur

- [1] M. Atzmüller, B. Klöpper, H.E. Al Mawla, B. Jäschke, M. Hollender, M. Graube, D. Arnu, A. Schmidt, S. Heinze, L. Schorer, A. Kroll, G. Stumme und L. Urbas. „Big Data Analytics for Proactive Industrial Decision Support: Approaches & First Experiences in the Context of the FEE Project“. In: *atp edition* 58. S. 62–74. 2016.
- [2] M.M. Breunig, H.P. Kriegel, R.T. Ng und J. Sander. „LOF: identifying density-based local outliers“. In: *ACM sigmod record* 29. S. 93–104. 2000.
- [3] M.A. Pimentel, D.A. Clifton, L. Clifton und L. Tarassenko. „A review of novelty detection“. In: *Signal Processing* 99. S. 215–249. 2014.
- [4] P. Esling und C. Agon. „Time series data mining“. In: *ACM Computing Surveys* 45. S. 1–34. 2012.
- [5] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang und E. Keogh. „Querying and mining of time series data: experimental comparison of representations and distance measures“. In: *Proceedings of the VLDB Endowment*. 2008.
- [6] E. Keogh, K. Chakrabarti, M. Pazzani und S. Mehrotra. „Locally adaptive dimensionality reduction for indexing large time series databases“. In: *ACM SIGMOD Record*. 30. 151–162. 2001.
- [7] J. Lin, E. Keogh, S. Lonardi und B. Chiu. „A symbolic representation of time series, with implications for streaming algorithms“. In: *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery* S. 2–11. 2003.
- [8] S. Chakraborty, S. Sarkar und A. Ray. „Symbolic identification for fault detection in aircraft gas turbine engines“. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 226. S. 422–436. 2012.
- [9] D. Francois, V. Wertz und M. Verleysen. „The concentration of fractional distances“. In: *IEEE Transactions on Knowledge and Data Engineering* 19. S. 873–886. 2007.

Reglerentwurf mit klassischen und Computational Intelligence Methoden in der Hochschullehre am Beispiel von Modellen

Tobias Loose

Hochschule Heilbronn, Technisches Logistikmanagement
Max-Planck-Str. 39, 74081 Heilbronn
E-Mail: tobias.loose@hs-heilbronn.de

Kurzfassung

Laboreinrichtungen bieten für die Hochschullehre eine wertvolle Ergänzung, um Vorlesungsinhalte sehr praxisnah umzusetzen. In diesem Beitrag wird anhand von kleinen Labormodellen demonstriert, wie hierbei Methoden der Regelungstechnik und der Computational Intelligence angewendet werden können. Es werden einige industriepraktische, teilweise komplexe Fragestellungen aufgezeigt, z. B. die Dämpfung von Schwingungen, der Umgang mit fehlender Kenntnis von Regelstrecken oder Streckenparametern, der Umgang mit schlecht regelbaren Systemen und die Platzierung von Sensoren. Diese Fragestellungen werden im Hinblick auf reale Anlagen diskutiert, um i. S. von Laborarbeiten den hohen Praxisbezug herzustellen. Dadurch ergänzen sie Lehrbücher, die oftmals lediglich idealisierte Modelle verwenden. Beispielsweise wird meistens die Rückführung der Regelgröße als ideal angenommen. Allerdings wird in realen Anwendungen nur eine *gemessene* Regelgröße zurück geführt, d. h. die Regelgröße ist u. a. nur mit systematischen oder zufälligen Messunsicherheiten, Verzögerungen, indirekten Schätzungen zugänglich. Eine bestimmte regelungstechnische Methode erweist sich aber erst dann als praxistauglich, wenn sie mit derartigen Schwierigkeiten erfolgreich umgehen kann.

1 Einführung

Bei der Ingenieurausbildung an Hochschulen werden oftmals Laborarbeiten zur Vertiefung von Vorlesungsinhalten für Studierende angeboten. Teilweise

kommen dabei reale Industrieanlagen zum Einsatz, was aber auch mit erheblichen (Anschaffungs-)Kosten und Betreuungsaufwand durch wissenschaftliche Mitarbeiter verbunden ist. Die Neuentwicklung, Realisierung und Inbetriebnahme einer *realen* Anlage durch Studierende kann dabei nicht erfolgen, was aber eine sehr gute Lernerfahrung wäre, denn „nur in der Praxis zeigt sich wie gut eine Theorie ist“. Um Aufwand und Kosten minimal zu halten, werden meistens Auszüge aus der Anlagenentwicklung als Laborarbeit angeboten, wie computergestützte Simulationen oder Versuche mit Komponenten wie z. B. Messdatenerfassung, siehe auch [21].

In diesem Beitrag wird als Ergänzung zu bestehenden Laborarbeiten der Einsatz von relativ kostengünstigen mechatronischen Modellen am Beispiel von LEGO® MINDSTORMS® vorgeschlagen. Sie bestehen aus einem Microcontroller, Sensoren, Mechanik und Antrieben, siehe Bild 1. Weitere derartige Modelle sind u. a. Arduino bzw. Genuino oder Raspberry Pi in Kombination mit klassischem Modellbau oder fischertechnik®, siehe auch [2, 5]. Bei den meisten der genannten Modellen lässt sich der Microcontroller an gängige Software wie MATLAB® / Simulink® anbinden, so dass eine Ingenieurwissenschaftliche Entwicklung einer kleinen mechatronischen Anlage möglich ist. Studierende erlangen dadurch einen Einblick in einen kompletten Entwicklungsprozess, d. h. von der Projektierung bis zur Inbetriebnahme, siehe auch [15]. Damit lassen sich viele Anlagen aus der Industriepraxis und typische Problemstellungen nachbilden, wie z. B. autonome Roboter, Regalbediengeräte oder Hubwerke und Probleme wie Kompensation von Störgrößen, Umgang mit nichtlinearen Elastizitäten, Ruckgleiten, Schlupf von Antriebsrädern, Materialtoleranzen, Messunsicherheiten oder Schwingungsreduzierung, siehe auch [14]. Diese Einflüsse sind durch geeignete Methoden auszugleichen, um eine akzeptable Regelgüte zu erreichen.



Bild 1: Überblick Komponenten, Bilder nach [24]

Dieser Beitrag stellt somit Ideen und Anregungen am Beispiel eines fahrerlosen Transportsystems zur Verfügung, um industriepraktische Anlagen in kleinen Modellen nachzubilden. Er liefert Anleitungen, wie Fragestellungen formuliert werden können, um im Rahmen von Laborarbeiten eingesetzt zu werden. Er soll zu weiteren Ideen zum Ausbau von Laborarbeiten anregen. Damit sollen Kompetenzen zur Beurteilung von *realen, gesamten* Anlagen aufgebaut werden, d. h. wie gut alle Komponenten wie Mechanik, Aktoren und Sensoren sowie alle Steuerungen und Regelungen zusammenarbeiten, um einen Prozess zu automatisieren. Es existieren zwar viele Lösungen zu einzelnen Problemstellungen, z. B. Regelung eines inversen Pendels (kommerziell auch unter „Segway“ bekannt), autonomes Fahren durch Labyrinth, autonomes Fahren entlang einer Linie, siehe [1, 9] und Internet-Recherche z. B. mit den Suchwörtern „smooth line follower ev3“. Im Rahmen einer Laborarbeit sollen allerdings *eigene* oder sogar verbesserte Lösungen erarbeitet werden, wie auch nachfolgend gezeigt wird. Zudem können viele Vor- und Nachteile verschiedener Methoden anschaulich demonstriert werden, wie in diesem Beitrag durch die Auslegung mit verschiedenen regelungstechnischen Methoden. Eine gewisse umfassende technische Erfahrung ist für viele ingenieurtechnische, industriepraktische Tätigkeiten eine solide Basis für einen erfolgreichen Berufsweg, siehe auch [16, 17, 20].

2 Labormodell eines fahrerlosen Transportsystems

Fahrerlose Transportsysteme (FTS) sind autonome Fahrsysteme, die berührungslos geführt werden. Die berührungslose Führung kann mit einer magnetisch-induktiven oder optischen Leitspur oder als freie Navigation mit Laser und Reflektoren an den Wänden realisiert werden, siehe [22, 23]. Nachfolgend wird eine optisch geführte Navigation beschrieben. Wie in der Einleitung erwähnt, existieren hierzu bereits mehrere Lösungen für LEGO® MINDSTORMS®, z. B. die Fuzzy-Regelung mit rein subjektiv eingestellten Zugehörigkeitsfunktionen, siehe [6]. Die Lösungen in diesem Beitrag sind verbesserte Lösungen, wie z. B. die nach einem Gütekriterium optimierte Fuzzy-Regelung. Zudem werden hier typische industriepraktische Überlegungen beim Entwurf der Regelung gezeigt, z. B. die Vor- und Nachteile eines klassischen PID- ggü. eines Fuzzy-Reglers.

Als Laborarbeit können am Beispiel des FTS folgende Aufgaben mit steigendem Schwierigkeitsgrad formuliert werden:

- *Rein reproduzierend:* Nachbau existierender Lösungen und Vergleich der Lösungen, z.B. die gezeigte Lösung aus diesem Beitrag (als rein reproduzierende Aufgabe ist das in einem ingenieurwissenschaftlichen Studium eher trivial).
- *Identifikation von Parametern:* Nachbau existierender Lösungen, zudem sollen durch Studierende geeignete Reglerparameter gefunden werden (entweder heuristisch oder mit entsprechender Modellbildung).
- *Komplette Entwicklung und Inbetriebnahme:* Es soll ein FTS entwickelt werden. Hierbei ist eine geeignete Mechanik zu entwerfen inkl. geeigneter Positionierung von Antrieb und Sensorik. Dabei hat die Positionierung der Antriebe Einfluss auf die Dynamik der Anlage, z. B. durch Wahl eines geeigneten Getriebefaktors entlang der kinematischen Kette. Die Positionierung der Sensoren hat Einfluss auf die gemessene Sensor-Kennlinie oder die Qualität der Messung (direkte/indirekte Messung von Größen, Einfluss von Störgrößen). Zudem ist ein geeigneter bzw. sind mehrere geeignete Regler zu entwerfen und zu implementieren.
- *Wettbewerb:* Ein FTS soll entwickelt werden, um einen vorher festgelegten Parcours so schnell wie möglich zu durchfahren. Hierbei treten Studierende (in Gruppen) gegeneinander an. Im Rahmen einer Laborarbeit ist diese Variante allerdings sehr zeitaufwändig.

2.1 Beispiel eines mechanischen Aufbaus

Im Rahmen dieses Beitrags wurde eine relativ einfache Mechanik als Fahrzeug realisiert, siehe Bild 2. Der Fokus des Beitrags liegt auf dem Vergleich von Regelungsstrategien und nicht auf der Entwicklung der Mechanik. Im Fahrzeug wurden zwei Antriebsmotoren jeweils für die linke und rechte Seite des Fahrzeugs eingebaut. Ein Antrieb treibt direkt über eine Welle ein Rad an. Dadurch ist das Fahrzeug sehr wendig und somit bei Kurvenfahrten sehr dynamisch. Bei Geradeausfahrt ist allerdings die Geschwindigkeit auf die maximal mögliche Motordrehzahl beschränkt und nicht weiter über einen Getriebefaktor angehoben. Zudem wurde ein Heckantrieb realisiert, um eine sehr gute Beschleunigung aus dem Stand ohne durchdrehende Räder zu erreichen. Der vordere Fahrzeugteil ist über ein Kugelrad abgestützt, daher sind auch Kurvenfahrten um die eigene Hochachse möglich, was später mit der klassischen PID-Regelung und sehr hohen Geschwindigkeiten gezeigt

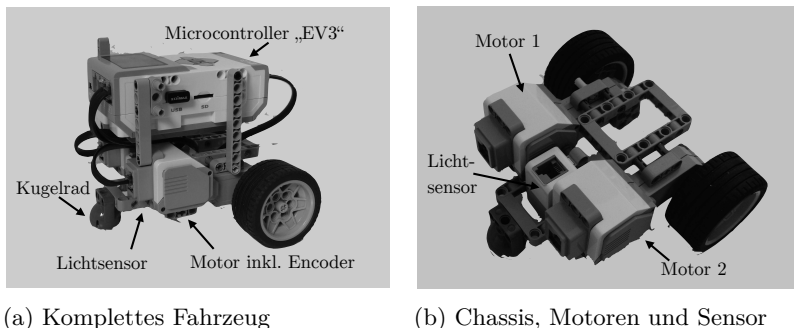


Bild 2: Aufbau des Fahrzeugs

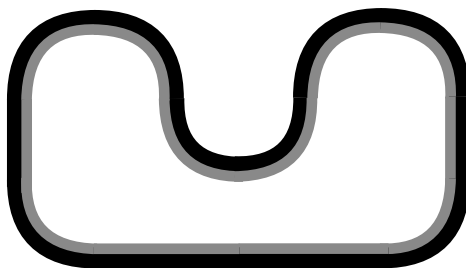


Bild 3: Beispiel eines Parcours zur Führung des Fahrzeugs

wird. Der Nachteil bei drei Stützpunkten (zwei Antriebsräder und ein Kugelrad) ist die Gefahr des Kippens bei Kurvenfahrt und bei Beladung, weil der Schwerpunkt außerhalb der drei Stützpunkte liegen kann.

2.2 Sensorplatzierung und -kennlinie

Die berührungslose Führung des Fahrzeugs wird hier durch einen optischen Sensor realisiert, der die Helligkeit eines reflektierten Lichtes vom Untergrund des Fahrzeuges misst. Durch eine schwarze bzw. mit Graustufen abgestufte Bahn auf weißem Untergrund lässt sich somit ein Parcours gestalten, siehe Bild 3. Entlang des Parcours soll das Fahrzeug geführt werden.

Selbstverständlich lassen sich verschiedene Bahnen mit unterschiedlichen Grau-Abstufungen realisieren, die jeweils eine Sensor-Kennlinie bedingen,

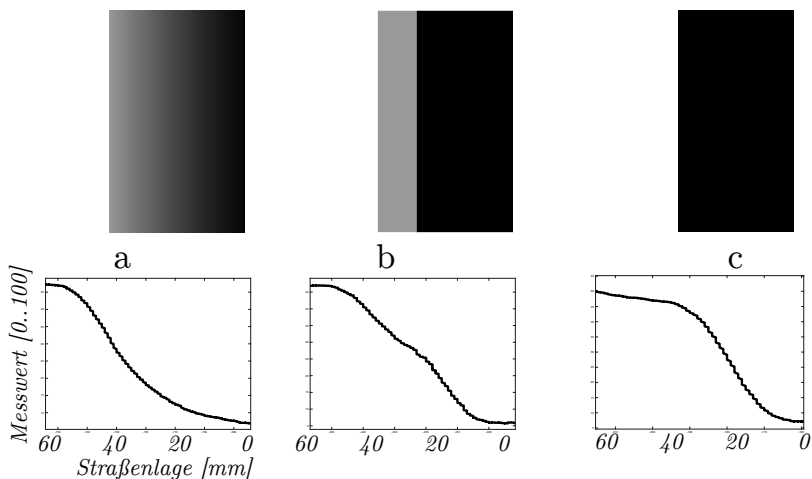


Bild 4: Beispiele für Bahnen zur optischen Führung des Fahrzeugs (oben) und deren Sensor-Kennlinien zur Bestimmung der Straßenlage (unten)

siehe Bild 4. Die Kennlinie gibt die Position des Fahrzeugs als „Straßenlage“ wieder. Je nach Gestaltung der Bahn durch z.B. Grau-Abstufungen, Bahnbreite oder Abstand des Sensors zur Bahn wird eine mehr oder weniger „gute“ Kennlinie erzielt. Hierbei bedeutet „gut“ im Sinne einer späteren erzielbaren Regelgüte. Üblicherweise sind lineare Kennlinien, minimale Störeinflüsse usw. zu bevorzugen. Nachfolgend wird bewusst eine „nicht so gute“ Kennlinie ausgewählt, um die Leistungsfähigkeit verschiedener Regler zu vergleichen. In der Industriepraxis ist ein robuster Regler unabdingbar, der trotz gewisser Nichtlinearitäten, Störeinflüssen, Schwankungen von Streckenparametern usw. eine gute Regelgüte liefert.

2.3 Klassischer PID-Regler

Sehr viele Regler werden in der Industriepraxis immer noch heuristisch eingestellt, in Anlehnung an die bereits lange bekannten Verfahren von Ziegler-Nichols, siehe [7, 19]. Die Gründe für diese Art der Regler-Einstellung sind u. a.

- die gute Interpretierbarkeit der Regler-Parameter (die einzelnen Parameter stellen bestimmtes Anlagenverhalten ein, z.B. die Schnelligkeit oder Genauigkeit),
- die einfache, in relativ kurzer Zeit realisierbare Implementierung bei einer oftmals guten / akzeptablen Regelgüte (die Parameter können während des Betriebs online eingestellt werden und das Ergebnis wird sofort sichtbar, was auch in der Industriepraxis ausgenutzt wird),
- die „ausgewogene“ Einstellung der Parameter unter Berücksichtigung vieler Aspekte (es können noch weitere Anforderungen an die Maschine berücksichtigt werden, z.B. robuster Regler bei unterschiedlichen Geschwindigkeiten; teilweise müssen bei anderen Verfahren alle Anforderungen explizit eingeplant werden, z. B. bei Neuronalen Netzen),
- die Möglichkeit einer (Nach-)Optimierung der Regelgüte an der Maschine trotz fehlendem bzw. nicht genauem mathematischem Modell (Änderungen der Reglerparameter sind an der realen Anlage sofort erkennbar und messbar).

Diese Gründe lassen sich im Rahmen einer Laborarbeit mit LEGO® MINDSTORMS® sehr gut nachvollziehen, da hier ebenso eine „reale“ Anlage in Betrieb genommen wird, und die Ergebnisse sind an der Anlage sofort zu sehen.

Die Stellgröße y ist hier der Lenkwinkel, und den Motoren wird eine Geschwindigkeit v_a zusätzlich aufgeprägt, siehe Bild 5. Dadurch lässt sich der Regelkreis auch auf andere Fahrzeuge anwenden, z.B. Fahrzeuge mit *einem* Antriebsmotor und einem Lenkmotor. Die tatsächlich resultierende translatorische Fahrzeuggeschwindigkeit wird mit

$$v_m = \frac{|v_1| + |v_2| - |v_1 - v_2|}{2} \quad (1)$$

berechnet, wobei v_1 bzw. v_2 die gemessene, translatorische Geschwindigkeit der jeweiligen Motoren ist. Die Motoren haben eingebaute Encoder, so dass die Motor-Winkelgeschwindigkeit erfasst und damit auch die translatorische Geschwindigkeit bestimmt werden kann. Beispielsweise wird eine Rotation des Fahrzeugs um die Hochachse mit $v_1 = +0,5 \text{ m/s}$ und $v_2 = -0,5 \text{ m/s}$ erreicht, was aber eine translatorische Bewegung mit $v_m = 0$ bedeutet, also keine Vorwärtsbewegung des Fahrzeugs.

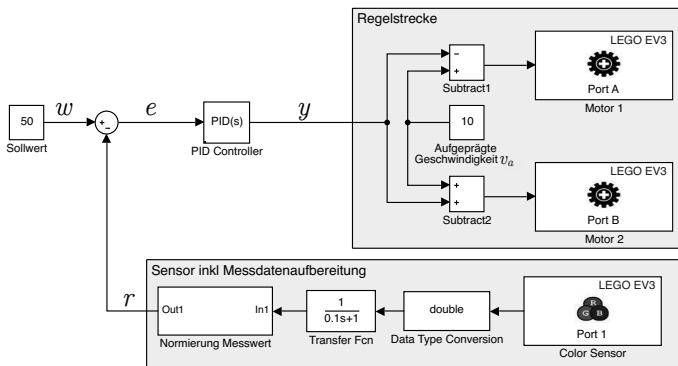


Bild 5: Realisierung des PID-Reglers durch Simulink[®] mit der Messgröße r , Sollwert w , Regeldifferenz e und Stellgröße y sowie der aufgeprägten Geschwindigkeit v_a . Durch die „Normierung des Messwertes“ werden Offsets ausgeglichen und es wird ein Wertebereich zwischen 0 und 100 ausgegeben

Zur Beurteilung der Regelgüte fährt das Fahrzeug entlang eines bestimmten Parcours. Dabei wird ein quadratisches Gütekriterium

$$\sigma_e = \sqrt{\sum_{t=1}^T [w(t) - r(t)]^2} = \sqrt{\sum_{t=1}^T e^2(t)} \quad (2)$$

mit dem Sollwert w , der Messgröße r und der Regeldifferenz e über eine bestimmte Zeit T verwendet. Damit wird auch das Schwingen um den Sollwert quantitativ erfasst. Zur Beurteilung der Regelgüte werden nachfolgend σ_e (soll möglichst klein sein) und v_m (soll möglichst groß sein) als Maße herangezogen, siehe Tabelle 1.

Die Reglerparameter lassen sich durch unterschiedliche Vorgehensweisen einstellen:

1. Eine rein heuristische Einstellung kann in diesem Beispiel mit einem reinen P-Regler beginnen. Eine Größenordnung für den P-Anteil K_P lässt sich aus der maximalen möglichen Regeldifferenz Δe_{max} (=entspricht üblicherweise des nahezu kompletten Sensor-Messbereichs) und den maximal möglichen Steuersignalen Δy_{max} mit $K_{P,MaxStart} = \frac{\Delta y_{max}}{\Delta e_{max}}$ ermitteln, siehe Tabelle 1 „Probelauf“. Zu erkennen sind sehr große Schwingungen bzw. sehr große Werte

Tabelle 1: Ergebnisse unterschiedlicher Einstellmethoden eines PID-Reglers bei verschieden aufgeprägten Geschwindigkeiten v_a , der tatsächlich erreichten Fahrzeug-Schwerpunktgeschwindigkeit v_m sowie der Standardabweichung der Regeldifferenz σ_e

Einstellmethode	Parameter				v_a	v_m	σ_e
	K_P	K_I	K_D	N			
Probelauf 1 mit $K_P = 0,5 \cdot K_{P,Start}$	0,5	0	0	0	10	5,01	11,03
					20	10,70	20,79
					30	instabil	instabil
Probelauf 2 mit $K_P = 1 \cdot K_{P,Start}$	1	0	0	0	10	1,01	29,84
					20	instabil	instabil
Heuristisch	0,8	0,4	0,06	80	10	4,82	5,38
					20	10,98	11,97
					30	14,37	20,73
Ziegler-Nichols	0,69	1,84	0,06	80	10	4,90	4,06
					20	11,97	11,71
					30	instabil	instabil
Matlab Autotuner	0,86	0,36	0,07	48	10	4,93	5,50
					20	11,23	10,75
					30	11,93	21,18

für σ_e . Als Lernerfahrung im Rahmen einer Laborarbeit lässt sich damit die Herangehensweise bei *realen* Industrieanlagen ableiten: In der Industriepaxis wird bei der Erstinbetriebnahme eher mit noch kleineren P-Anteilen begonnen, um sie sukzessive zu erhöhen. Es wird sich „langsam und mit Respekt vor teuren Maschinen herangetastet“, um nicht sofort an die Stabilitätsgrenze zu gelangen bzw. von ihr fern zu bleiben. Beispielsweise wird je nach Anlage mit $K_P = 0,1 \dots 0,5 \cdot K_{P,Start}$ begonnen. Danach werden sukzessive der I-Anteil (hier für die stationäre Genauigkeit) und D-Anteil zusammen mit Filter-Koeffizient des realen D-Anteils (hier für die dämpfende Wirkung des gesamten Regelkreises) erhöht, bis eine akzeptable Regelgüte erreicht wird. Wichtig ist dabei auch die geeignete Filterung des Sensor-Signals, da sonst insbesondere durch den D-Anteil die Anlage zum Schwingen neigt. Der Filter für den Sensor wurde hier pragmatisch mit einem P-T₁ Glied und einer Verzögerungszeit $T = 0,1$

realisiert, was hier ein guter Kompromiss aus Nutz-Störsignal und minimaler Phasenverschiebung ist.

In diesem Beispiel erreichen $K_P = 0,8$ sowie $K_I = 0,4$ und $K_D = 0,06$ mit $N = 80$ gute Ergebnisse bei unterschiedlich aufgeprägten Geschwindigkeiten v_a . Dieses gute Ergebnis ist durch hohe tatsächliche Geschwindigkeiten v_m und geringe Streuung der Regeldifferenz σ_e (=Maß für Schwingungen) über verschieden aufgeprägte Geschwindigkeiten v_a gekennzeichnet. Nachfolgend gezeigte Reglereinstellungen werden anhand dieser heuristischen Einstellung als Referenz gemessen. Weitere Optimierungen des Ergebnisses lassen sich u. a. durch Verbreiterung der Bahn oder Verwendung von Graustufen in der Bahn erzielen.

Bei LEGO[®] MINDSTORMS[®] in Verbindung mit MATLAB[®] / Simulink[®] lassen sich die Regleparameter online ändern, d. h. während dem Betrieb. Das hat für eine Laborarbeit einen hohen didaktischen Wert, da die Auswirkungen von Parameter-Änderungen sofort ersichtlich werden. Auch in der Industriepraxis werden bei der Inbetriebnahme die Regler-Parameter online optimiert. Dabei haben erfahrene Inbetriebnehmer „immer eine Hand am Not-Aus“, um bei drohender Instabilität oder drohendem Betrieb außerhalb der Grenzwerte die Anlage zu stoppen.

2. Für die Auslegung nach Ziegler-Nichols wird die Anlage mit $v_a = 0$ und einem reinen P-Regler bis an die Stabilitätsgrenze angeregt, so dass Dauerschwingungen auftreten. In diesem Beispiel ist das für $K_P = K_{P,krit} = 1,15$ erreicht, dabei wurde eine Schwingungsperiode von $T_{krit} = 0,75$ gemessen. Nach dem Verfahren von Ziegler-Nichols werden für einen PID-Regler die einzelnen Anteile zu $K_P = 0,6 \cdot K_{P,krit} = 0,69$ und $K_I = 1,2 \cdot K_{P,krit}/T_{krit} = 1,84$ sowie $K_D = 0,075 \cdot K_{P,krit} \cdot T_{krit} = 0,06$ bestimmt, siehe auch [19]. Der Filter-Koeffizient des realen D-Anteils wurde aus der heuristischen Methode mit $N = 80$ übernommen. Interessant bei den Ergebnissen ist, dass hier die Regler-Parameter bis auf den I-Anteil sehr nahe an der heuristischen Lösung liegen, siehe Tabelle 1. Ebenso zeigt sich für langsame Geschwindigkeiten mit $v_a = 10 \dots 20$ eine bessere Regelgüte als die heuristische Einstellung. Allerdings wird das Fahrzeug bei höheren Geschwindigkeiten instabil und kann den Kurven der Parcours-Bahn nicht mehr folgen.

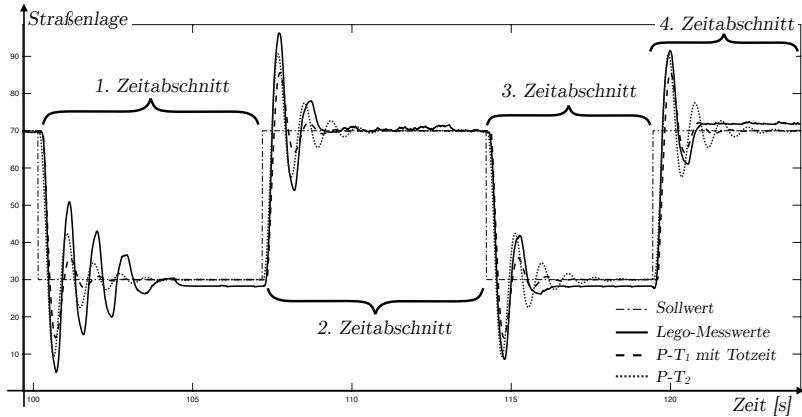


Bild 6: Vergleich realer Messwerte mit zwei Modellansätzen

3. Der Regler lässt sich natürlich auch mit einer vorgelagerten Modellbildung entwerfen. Anhand der realen LEGO[®] Anlage lässt sich allerdings auch die Schwierigkeit der Modellbildung demonstrieren. Besondere Schwierigkeiten resultieren hier u. a. durch die relativ hohen Abtastzeiten des Microcontrollers mit $t_{Abtast} = 0,1 \text{ s}$, Materialtoleranzen und -spiel, die als Hystereseeffekte und Nichtlinearitäten bei der Vermessung der Anlage auftreten. Beispielsweise hat hier ein Antriebsrad ein Spiel von $3^\circ \dots 4^\circ$ bedingt durch die Toleranzen der Getriebezahnräder. Das führt dazu, dass die Anlage teilweise unterschiedlich auf die gleiche Anregungsfunktion reagiert, vergleiche 1. und 3. Zeitabschnitt mit 8 bzw. 3 Schwingungsaussschlägen der Lego-Anlage in Bild 6. Derartige Phänomene lassen sich nur schwer bzw. gar nicht mit Modellen nachbilden, da sich z. B. aktuelle Zahnradspiele nicht vorhersagen lassen. Dennoch lässt sich mit den (unvollständigen) Modellen ein Regler auslegen, entsprechend dem berühmten Zitat „Essentially, all models are wrong, but some are useful“, siehe S. 424 in [3]. Diese Effekte sind auch bei realen Industrieanlagen vorhanden, allerdings in einer anderen Größenordnung, aber es sind natürlich auch höhere Anforderungen an die Regelgüte gestellt, siehe beispielsweise [8, 13]. Die Anlage kann durch ein integrierendes Verzögerungsglied 2. Ordnung (P-T₂)

$$F_{M1}(s) = \frac{K_1 \cdot \omega_0}{s^2 + 2 \cdot D \cdot \omega_0 \cdot s + \omega_0^2} \cdot \frac{1}{s} \quad (3)$$

approximiert werden, siehe Bild 6 und 7a. Der theoretische Hintergrund ist das Modell eines Elektromotors, der eine Masse antreibt mit der Spannung als Eingangs- und Position als Ausgangsgröße, siehe auch [7, 10, 14]. Die Parameter können mit Hilfe von Messwerten oder mit Hilfe von Optimierungsalgorithmen bestimmt werden. Für dieses Beispiel wird mit den Parametern Eigenkreisfrequenz des ungedämpften Systems $\omega_0 = 10 \text{ s}^{-1}$, Dämpfungsgrad $D = 0,8$ und Verstärkungsfaktor $K_1 = 8$ eine akzeptable Modellgüte erreicht, siehe punktierte Linie in Bild 6. Eine weitere Möglichkeit die Anlage zu modellieren besteht in der Annahme einer Totzeit T_t mit einem integrierenden Verzögerungsglied 1. Ordnung (P-T₁)

$$F_{M2}(s) = \frac{K_2}{T \cdot s + 1} \cdot \frac{1}{s} \cdot e^{-T_t \cdot s}, \quad (4)$$

siehe Bild 6 und 7b. Der theoretische Hintergrund ist das vereinfachte Modell eines Elektromotors, der eine Masse antreibt. Das Spiel bzw. die Materialtoleranzen sowie Modellfehler sind durch die Totzeit T_t abgedeckt. Durch Vergleich mit realen Messwerten konnte mit den Parametern Totzeit $T_t = 0,15 \text{ s}$, Verstärkungsfaktor $K = 5$ und Verzögerungszeit $T_s = 0,02$ eine gute Approximation an das reale Verhalten erzielt werden, siehe gestrichelte Linie in Bild 6.

Auf Basis derartiger Modelle kann der Regler u. a. mit der MATLAB[®] / Simulink[®] Funktion „Autotuner“ parametrisiert werden, beispielsweise wurde mit „Modell 2“ die Parameter $K_P = 0,86$ und $K_I = 0,36$ sowie $K_D = 0,07$ mit $N = 48$ gefunden, siehe auch Tabelle 1. Interessant bei dem Ergebnis ist, dass die Parameter sehr ähnlich zu den heuristisch gefundenen Werten sind. Dies zeigt, dass für die Industriepraxis akzeptable Werte durch heuristische Einstellung bei bestimmten Anlagen gefunden werden können. Bei komplexeren Industrieanlagen wird nach Möglichkeit im Vorfeld ein Wertebereich von Parametern durch Simulation der Anlage identifiziert, die Optimierung erfolgt aber auch dann i. d. R. vor Ort an der Maschine.

4. Viele weitere Einstellmethoden oder Erweiterungen des Regelkreises lassen sich für Laborarbeiten implementieren, wie z.B. Kompensation von Zeitkonstanten, Kaskadenregelung mit einem unterlagerten Geschwindigkeitsregler oder die Aufschaltung von Störgrößen.

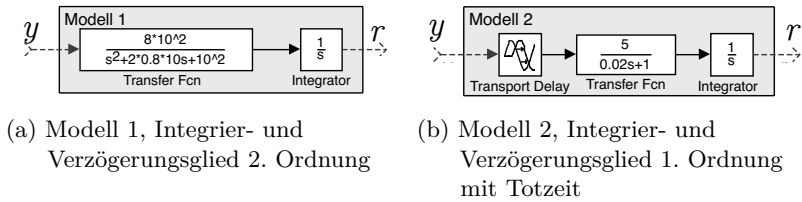


Bild 7: Simulink[®] Modelle mit dem Lenkwinkel y als Eingangs- und der Straßenlage r als Ausgangsgröße

2.4 Fuzzy-Regler mit optimalen Zugehörigkeitsfunktionen

Ein weiterer Ansatz die LEGO[®] MINDSTORMS[®] Anlage zu regeln, besteht in der Verwendung eines Fuzzy-Reglers. Hierfür bietet MATLAB[®] / Simulink[®] vorgefertigte Toolboxes. Auch hier lassen sich die Parameter online einstellen, so dass die LEGO[®] Anlage während dem Betrieb noch optimiert werden kann.

In diesem Beispiel wird zunächst die Position des FTS fuzzifiziert und über die Regeln

WENN Straßenlage = Mitte DANN Lenkung = Mitte

WENN Straßenlage = rechts DANN Lenkung = links

WENN Straßenlage = links DANN Lenkung = rechts

der Lenkwinkel ausgegeben, siehe „Fuzzy Logic Controller 1“ in Bild 8.

Die Defuzzifizierung der Lenkung bzw. Lenkwinkels wird hier mit der Schwerpunktmethod berechnet und als Stellgröße y ausgegeben. Die Werte der Stützpunkte der Zugehörigkeitsfunktionen (ZGF) wurden zunächst willkürlich gewählt, aber mit einer gewissen Plausibilität. Aufgrund der Einfachheit wurden zunächst mit so wenig wie möglichen Stützpunkten die ZGF gestaltet, d. h. das Minimum einer ZGF fällt mit dem Maximum der nächsten ZGF zusammen, siehe Bild 9a.

Mit dem Fuzzy-Regler konnte die Anlage schnell und plausibel i. S. einfacher, einzeln interpretierbarer Regeln in Betrieb genommen werden, d. h. die Anlage funktioniert in einem stabilen Zustand. Im Rahmen einer Laborarbeit wird der Vorteil des Fuzzy-Reglers ersichtlich, indem ohne genaue physikalische Prozesskenntnis ein funktionierender Regler entworfen werden kann. Bei einem PID-Regler muss zumindest die Modellstruktur der Regelstrecke in etwa bekannt sein, um von vornherein eine Instabilität zu vermeiden. Daher kommen auch bei Industrieanlagen, bei denen der zu re-

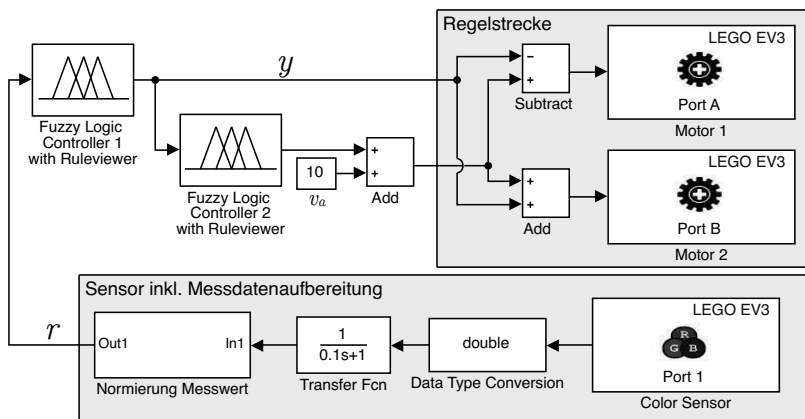


Bild 8: Realisierung des Fuzzy-Reglers mit der Simulink[®] Toolbox „Fuzzy Logic Controller with Ruleviewer“

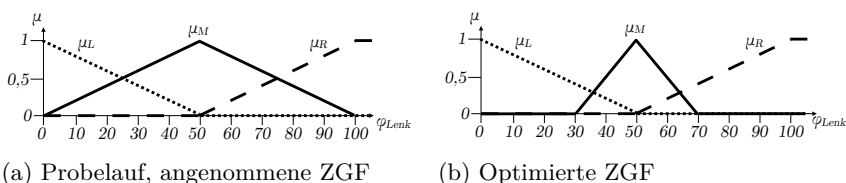


Bild 9: ZGF für den Lenkwinkel mit μ_L = links, μ_M = Mitte, μ_R = rechts

gelnde Prozess sehr schwer zu modellieren bzw. nur ansatzweise bekannt ist oder viele nichtlineare Einflussgrößen zu berücksichtigen sind, Fuzzy-Regler zum Einsatz, siehe [11].

In diesem Beispiel zeigt sich allerdings auch, dass die Regelgüte des Fuzzy-Reglers zunächst schlecht ist, sie ist vergleichbar mit dem „Probelauf“ bei der Einstellung des PID-Reglers, siehe Tabelle 1 und 2. Dies führt wiederum zu einem Optimierungsproblem, um geeignete Regeln und Parameter zu finden. Im Vergleich zum PID-Regler existieren aber beim Fuzzy-Regler eine Vielzahl von Einflussgrößen, z. B. die Wahl von Stützpunkten der ZGF, Anzahl von ZGF, Anzahl von Regeln, Auswahl einer Methode zur Defuzzifizierung. Zwar sind *einzelne* Fuzzy-Regeln oder *einzelne* ZGF interpretierbar, allerdings ist eine Vielzahl von Regeln und ZGF nicht mehr interpretierbar, insbesondere hinsichtlich des Einflusses auf Stabilität der Anlage, Schwingungsreduzierung usw. Das bedeutet, dass eine heuristische

Tabelle 2: Ergebnisse des Fuzzy-Reglers, „instabil“ bedeutet hier ein Herausfahren aus der Kurve, d. h. der Regler ist zu langsam

Einstellmethode	v_a	v_m	σ_e
Fuzzy „Probelauf“	10	4,95	12,40
	20	instabil	instabil
Fuzzy, optimiert	-	8,85	12,18

Inbetriebnahme mit einer online Optimierung schwer möglich ist, weil die Vielzahl an Parametern schwer überschaubar ist. Auch in anderen Bereichen, wie z. B. das Investieren in eine *Vielzahl* von Aktien, führt eine große Menge an Variablen zu einem nur noch schwer überschaubarem Gesamtbild, siehe [4, 12]. Bei Fuzzy-Reglern werden üblicherweise Optimierungsalgorithmen verwendet, um automatisiert nach geeigneten Regler-Parametern zu suchen, siehe [11, 18].

Zur Optimierung lassen sich viele Methoden anwenden, z. B. das Gradientenabstiegsverfahren. Zuvor müssen geeignete Parameter zur Variation festgelegt werden, z. B. die Stützpunkte der ZGF. In diesem Beispiel sind zudem noch weitere Regeln ergänzt worden, um die Geschwindigkeit beim Kurvenfahren zu reduzieren, siehe „Fuzzy Logic Controller 2“ in Bild 8 bei gleichzeitigem „Ausschalten“ einer vorgeprägten Geschwindigkeit mit $v_a = 0$. Diese Regelbasis beinhaltet

WENN Straßenlage = Mitte DANN Geschwindigkeit = hoch sowie
WENN Straßenlage \neq Mitte DANN Geschwindigkeit = niedrig.

Diese Regeln erscheinen in Anlehnung an das Autofahren sinnvoll, zumal der Fuzzy-Regler aus der Kurve heraus fährt, wie oben beschrieben.

Im Rahmen dieser Laborarbeit ist ersichtlich, dass viele Iterationsschritte notwendig sind, um ein Optimum zu finden. Dazu muss das Fahrzeug pro Iterationsschritt eine gewisse Strecke zurücklegen, um einen verlässlichen Wert des Gütekriteriums σ_e zu erhalten. Im Vergleich zur heuristischen Einstellung eines PID-Reglers reicht meistens die Beurteilung eines Einschwingvorganges aus, um die Regler-Parameter sinnvoll zu optimieren. In dieser Arbeit wird exemplarisch die Optimierung von Stützpunkten bei zwei ZGF mit dem Lenkwinkel μ_M und der aufgeprägten Geschwindigkeit μ_{va} gezeigt. Das Ergebnis zeigt, dass es optimale Stützpunkte für die ZGF des Lenkwinkels gibt, siehe Bild 10. Interessant ist, dass die Regeln für die Geschwindigkeiten „Fuzzy Logic Controller 2“ die Regelgüte verbessern,

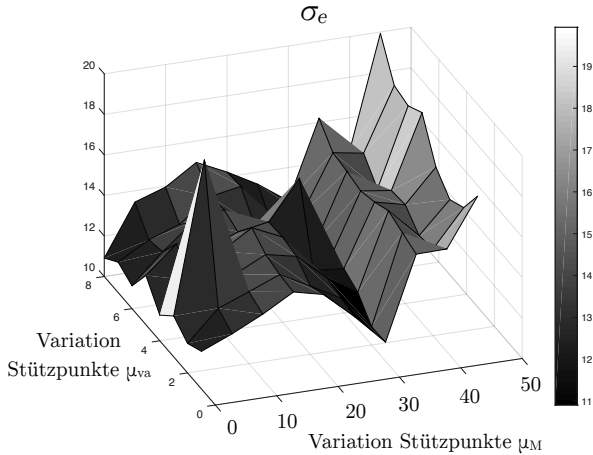


Bild 10: Gütegebirge für die Streuung der Regeldifferenz σ_e bei Variation von Stützpunkten der ZGF des Lenkwinkels μ_M und der Geschwindigkeit μ_{va}

d. h. dadurch sind höhere Geschwindigkeiten möglich ohne „aus der Kurve zu fahren“. Hingegen hat die Wahl von *Stützpunkten* der ZGF μ_{va} kaum Einfluss auf die Regelgüte. Dies zeigt wie komplex ein Fuzzy-Regler werden kann. Die tatsächlich erreichte Geschwindigkeit v_m hängt i.W. mit den Stützpunkten von μ_M zusammen, wobei ab den Stützpunkten [30, 50, 70] kaum nennenswerter Geschwindigkeitszuwachs statt findet, siehe Stützpunkte in Bild 9b und Ergebnis in Bild 11.

Die Auslegung eines guten Fuzzy-Reglers ist in diesem Beispiel sehr viel aufwändiger im Vergleich zu einem PID-Regler. Um auf ähnliche Ergebnisse wie die vom PID-Regler zu kommen, müssen weitere Regeln implementiert werden, wie z. B. der Einbezug der *Änderung* der Straßenlage, um die Wirkung eines D-Reglers nachzubilden. Zudem sind die Regeln *im Nachhinein*, d. h. nach einer computergestützten Optimierung, sehr gut interpretierbar, um ein Verständnis für den zugrundeliegenden Prozess zu erlangen, siehe [11, 18]. Beispielsweise führt die optimierte ZGF μ_M mit den Stützpunkten [30, 50, 70] zu einem eher „aggressiven“ Lenkverhalten, d. h. die Lenkbewegung wird früher eingeleitet als mit den Stützpunkten [0, 50, 100].

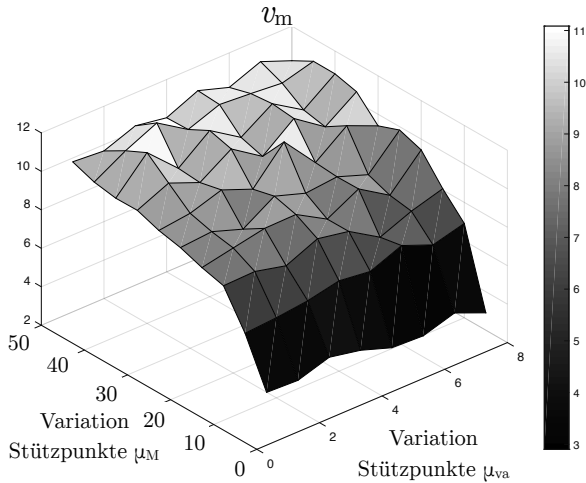


Bild 11: Gütegebirge für die erreichte Geschwindigkeit v_m bei Variation von Stützpunkten der ZGF des Lenkwinkels μ_M und der Geschwindigkeit μ_{va}

2.5 Ausblick: Zustandsregler und Regelung mit Neuronalen Netzen

In dem hier gezeigten Stil lassen sich noch weitere Methoden zur Regelung von Anlagen als Laborarbeit durchführen, was nachfolgend nur kurz erläutert wird. Es lässt sich auf Basis des Modells von Gleichung (3) ein Zustandsregler entwerfen, eine ausführliche Anleitung ist in [14] beschrieben. Im Rahmen einer Laborarbeit lassen sich die Vorteile des Zustandsreglers sehr gut demonstrieren, wie z. B. das Bestimmen des Anlagenverhaltens durch die Pol-Vorgabe. Auf der anderen Seite ist eine genaue Modellbeschreibung der Anlage notwendig. Wie oben gezeigt, ist genau das insbesondere bei der LEGO[®] Anlage durch die vielen Toleranzen und Nichtlinearitäten eine gewisse Herausforderung. Zudem ist eine online-Nachjustierung des Zustandsreglers eher schwierig, was aber für Industrieanlagen vor Ort bei der Inbetriebnahme sehr wichtig ist.

Des Weiteren lässt sich auch ein Regler durch Neuronale Netze anlernen. Das Anlernen kann entweder sehr aufwändig online mit der LEGO[®] Anlage erfolgen oder computergestützt anhand von Modellen, wie mit Gleichung (3) und den diskutierten Schwierigkeiten der Modellgüte und der fehlenden Möglichkeit einer heuristischen online-Nachjustierung.

2.6 Anregung für weitere Anlagen

Nachfolgend sind noch Ideen genannt, die als Laborübungen in Anlehnung an industriepraktische Anlagen genutzt werden können, d. h. mit LEGO[®] MINDSTORMS[®] realisierbar sind, z. B.

- Regalbediengerät: Positionsregelung des Fahrzeugs bei gleichzeitiger Schwingungsreduzierung des Masts, siehe auch [14],
- Pendeldämpfung: Positionsregelung einer Kran-Laufkatze bei gleichzeitiger Schwingungsreduzierung der hängenden Last am Seil,
- Einparkassistent mit / ohne Anhänger: Automatisches Einparken eines Fahrzeugs mit / ohne Anhänger in eine Parklücke,
- Fluidtechnische Hubbühne: Ventilgeregelte Positionierung einer pneumatischen bzw. hydraulischen Anlage, siehe auch [15],
- Regelung einer Hexapod Plattform mit mehreren Antrieben,
- Gleichlaufregelung einer Werkzeugmaschine, z.B. eine Schere,
- Industrieroboter: Regelung eines Mehrgelenk-Roboters entlang einer vorgegebenen Trajektorie,
- Tiefziehpresse: Positionsregelung eines Pressenstößels und Kraftregelung des Niederhalters, siehe [13].

2.7 Zusammenfassung

In dieser Arbeit wurden kleine physische, mechatronische Modelle als Ergänzung zu Laborarbeiten in der Hochschullehre vorgeschlagen. An einem konkreten Beispiel eines fahrerlosen Transportsystems (FTS) wurde die Entwicklung und Implementierung verschiedener Reglerstrategien mit LEGO[®] MINDSTORMS[®] aufgezeigt und im Kontext der Industriepaxis diskutiert. Durch die gezeigte Diskussion lassen sich die Vor- und Nachteile verschiedener Regler und deren Auslegungsmethoden im Rahmen einer Laborübung sehr anschaulich demonstrieren. Studierende erfahren zudem die Inbetriebnahme einer realen physischen Anlage.

Schwerpunkt dieser Arbeit war der industriepraktische Vergleich verschiedener Regler und deren Parameteridentifikation. Es wurden Gründe gezeigt, warum die heuristische Auslegung eines PID-Reglers nach wie vor bei vielen Industrieanlagen angewandt wird, u. a. wegen der online-Optimierung „vor

Ort“. Die Vorteile des Fuzzy-Reglers wurden ebenso gezeigt, wie die Inbetriebnahme ohne genaue Modellkenntnisse. Hingegen ist die Vielzahl der Einstellmöglichkeiten eine Schwierigkeit des Fuzzy-Reglers und es besteht daher Bedarf an rechnergestützten Methoden, um einen optimalen Fuzzy-Regler zu erhalten. Nach einer rechnergestützten Optimierung ist wiederum die Interpretation der gefundenen Regeln und Zugehörigkeitsfunktionen mit den Stützpunkten sehr interessant, um Prozesskenntnisse zu gewinnen.

Literatur

- [1] A. Behrens, L. Atorf, R. Schwann, B. Neumann, R. Schnitzler, J. Ballé, T. Herold, A. Telle, T. Noll, K. Hameyer, T. Aach. „MATLAB Meets LEGO Mindstorms—A Freshman Introduction Course Into Practical Engineering“. In: *IEEE Transactions on Education* Vol. 53, Iss. 2. S. 306 - 317. 2010.
- [2] C. Bell. „Beginning Sensor Networks with Arduino and Raspberry Pi“. New York: Springer. 2013.
- [3] G. Box, N. Draper. „Empirical Model-Building and Response Surfaces“. John Wiley & Sons. 1987.
- [4] M. Buffett, D. Clark. „Das Tao des Warren Buffett“. Kulmbach: Börsenbuchverlag. 2008.
- [5] K. Dembowski. „Raspberry Pi – Das technische Handbuch“. Wiesbaden: Springer. 2015.
- [6] U. Farooq, H. Arif, M. Amar, B. Tahir, M. Qasim, N. Ahmad. „Comparative Analysis of Fuzzy Logic and Neural Network Based Controllers for Line Tracking Task in Mobile Robots“. In: *Proc. IEEE Signal Processing Systems (ICSPS)*. S. 733-739. 2010.
- [7] O. Föllinger. „Regelungstechnik“. Heidelberg: Hüthig. 1994.
- [8] K.-H. Grote, J. Feldhusen (Hg.). „Dubbel: Taschenbuch für den Maschinenbau“. Berlin, Heidelberg, New York: Springer. 2014.
- [9] D.-W. Gu, P. Petkov, M. Konstantinov. „Robust Control Design with MATLAB®“. London: Springer. 2013.
- [10] R. Hagl. „Elektrische Antriebstechnik“. München: Carl Hanser. 2013.

- [11] J. Kacprzyk, W. Pedrycz. „Springer Handbook of Computational Intelligence“. Berlin, Heidelberg: Springer 2015.
- [12] A. Kostolany. „Die Kunst, über Geld nachzudenken“. Berlin: Ullstein. 2015.
- [13] T. Loose, H. Mehling, E. Holembowski, S. Arns, K. Kempf, H. Behl. „Im Ziehkissen liegt die Kraft“. In: *BLECH InForm* Nr. 5, S. 104 - 107. 2006.
- [14] T. Loose: „Benchmark-Anwendung zur Schwingungsanalyse und -dämpfung von Regalbediengeräten am Beispiel eines Labormodells“. In: *Proc., 25. Workshop Computational Intelligence* (Mikut, R.; Reischl, M., Hg.), S. 127–144. Universitätsverlag Karlsruhe. 2015.
- [15] T. Loose: „Projektierung mechatronischer Anlagen als Laborarbeit in der Hochschullehre am Beispiel LEGO[®] MINDSTORMS[®]“. In: *Proc., 2. IFToMM D-A-CH Konferenz*, S. 1–8 (Tagungsband auf CD-ROM). Universität Innsbruck. 2016.
- [16] H. Mell. „Spielregeln für Beruf und Karriere“. Berlin, Heidelberg: Springer. 2013.
- [17] H. Mell. „Erfolgreiche Karriereplanung“. Berlin, Heidelberg: Springer. 2014.
- [18] R. Mikut. „Data Mining in der Medizin und Medizintechnik“. Karlsruhe: Universitätsverlag Karlsruhe. 2008.
- [19] W. Oppelt. „Kleines Handbuch technischer Regelvorgänge“. Weinheim / Bergstr.: Verlag Chemie. 1972.
- [20] R. Saint-Nom, D. Jacoby. „Building the first steps into SP Research“. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* Vol. 5. S. 545-548. 2005.
- [21] H. Scherf. „Modellbildung und Simulation dynamischer Systeme“. München: Oldenbourg. 2010.
- [22] G. Ullrich. „Fahrerlose Transportsysteme“. Wiesbaden: Springer. 2014.
- [23] VDI 2510. „Fahrerlose Transportsysteme“. Berlin: Beuth. 2005.
- [24] <http://www.lego.com>. Lego Homepage.

Dieser Tagungsband enthält die Beiträge des 26. Workshops „Computational Intelligence“ des Fachausschusses 5.14 der VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA) und der Fachgruppe „Fuzzy-Systeme und Soft-Computing“ der Gesellschaft für Informatik (GI), der vom 24. – 25.11.2016 in Dortmund stattfindet.

Der GMA-Fachausschuss 5.14 „Computational Intelligence“ entstand 2005 aus den bisherigen Fachausschüssen „Neuronale Netze und Evolutionäre Algorithmen“ (FA 5.21) sowie „Fuzzy Control“ (FA 5.22). Der Workshop steht in der Tradition der bisherigen Fuzzy-Workshops, hat aber seinen Fokus in den letzten Jahren schrittweise erweitert.

Die Schwerpunkte sind Methoden, Anwendungen und Tools für

- Fuzzy-Systeme,
- Künstliche Neuronale Netze,
- Evolutionäre Algorithmen und
- Data-Mining-Verfahren

sowie der Methodenvergleich anhand von industriellen und Benchmark-Problemen.

Die Ergebnisse werden von Teilnehmern aus Hochschulen, Forschungseinrichtungen und der Industrie in einer offenen Atmosphäre intensiv diskutiert. Dabei ist es gute Tradition, auch neue Ansätze und Ideen bereits in einem frühen Entwicklungsstadium vorzustellen, in dem sie noch nicht vollständig ausgereift sind.

