Herausgeber

H. SCHULTE
F. HOFFMANN
R. MIKUT

Berlin | 21. – 22. November 2024

PROCEEDINGS **34. WORKSHOP** COMPUTATIONAL INTELLIGENCE

KIT Scientific Publishing

H. Schulte, F. Hoffmann, R. Mikut (Hrsg.)

Proceedings. 34. Workshop Computational Intelligence

Berlin, 21. – 22. November 2024

PROCEEDINGS **34. WORKSHOP** COMPUTATIONAL INTELLIGENCE

Berlin, 21. – 22. November 2024

Herausgegeben von
H. Schulte
F. Hoffmann
R. Mikut

KIT Scientific Publishing

Impressum

KIT Scientific Publishing

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

www.ksp.kit.edu

Print on Demand 2024 – Gedruckt auf FSC-zertifiziertem Papier

ISBN 978-3-7315-1388-9
DOI 10.5445/KSP/1000174544

Inhaltsverzeichnis

AutoWP: Automated wind power forecasts with limited computing resources using an ensemble of diverse wind power curves

Stefan Meisenbacher[1], Silas Aaron Selzer[2,3], Mehdi Dado[1], Maximilian Beichter[1], Tim Martin[1], Markus Zdrallek[2], Peter Bretschneider[3,4], Veit Hagenmeyer[1], Ralf Mikut[1]

E-Mail: ralf.mikut@kit.edu, [1]Karlsruhe Institute of Technology, Germany
[2]University of Wuppertal, Germany [3]Ilmenau University of Technology, Germany
[4]Fraunhofer IOSB, Germany

Abstract

Forecasting the locally distributed Wind Power (WP) generation is crucial for future energy systems, demanding scalable WP forecasting models to keep pace with the increasing number of smart grid applications. Therefore, we propose AutoWP, which is a weighted ensemble of WP curves that represent different site conditions. This representation is achieved by using a diverse set of WP curves from Original Equipment Manufacturers (OEMs) and optimizing their contribution to the weighted sum of curves using the least squares method. AutoWP is advantageous since physical limitations in WP generation are implicitly reflected in the considered WP curves, and the optimization to find the ensemble weights requires only a small amount of data and computational effort. Furthermore, AutoWP uses a rule-based and filter-based approach for training data cleaning to identify and filter samples representing abnormal operation due to shutdowns or partial load operation. It is shown on a real-world data set that AutoWP achieves competitive day-ahead forecasting performance against other methods based on WP curve modeling and outperforms the autoregressive Deep Learning (DL) method Temporal Fusion Transformer (TFT), which is based on the attention mechanism.

1 Introduction

The increasing generation capacity of decentral Wind Power (WP) turbines and PhotoVoltaic (PV) plants with weather-dependent and fluctuating generation poses new challenges for the stability of power grids, especially for frequency stability. However, grid bottlenecks occur not only at the transmission grid level but also at the distribution grid level, making smart grid applications indispensable in the future energy system, e. g., demand response to reduce peak-valley load differences [1], battery scheduling to optimize WP dispatch planning [2], and redispatch planning for predictive congestion management [3]. Since many smart grid applications require forecasts of decentralized WP generation, it is necessary to automate the design and operation of WP forecasting models to keep pace with the growing number of applications and the expansion of generation capacity. On the one hand, models achieving low forecasting errors are required that consider site-specific conditions like the wind direction and the mutual influence of the WP turbines (wake losses) in offshore wind farms. Because offshore wind turbines are established in a uniform terrain, the WP farm can be modeled holistically like in [4] using an autoregressive Deep Learning (DL) model. On the other hand, scalable forecasting models are required for decentrally located onshore WP turbines, where the terrain is heterogeneous (open fields, urban areas, hills, and forests). In such a case, the use of DL models is limited since they require a high computational effort for training the model to represent site-specific air turbulence caused by the terrain. Even though so-called WP curves are available for most turbines, which establish an empirical relationship between wind speed and WP generation, the WP curves provided by the Original Equipment Manufacturer (OEM) only apply to the site-specific conditions in which they were recorded.

Related work on site-specific WP curve modeling comprise parametric and non-parametric modeling approaches [5]. While the former is based on fitting an assumed mathematical expression [6] or using physical-inspired models [7], the latter comprises methods based on Machine Learning (ML), e. g., Support Vector Regression (SVR) [8–10], the MultiLayer Perceptron (MLP) [7, 9, 11, 12], and Decision Tree (DT)-based methods [8, 10, 13]. ML-based methods can consider further explanatory variables such as the atmospheric pressure [14], the air temperature [12, 14] and the wind direction [11–13]. A critical challenge, however, is inconsistent training data due to shutdowns or partial load operation. Such inconsistencies can be classified into i) steady WP generation at a power below the peak power rating of the turbine, ii) no WP generation while above cut-in wind speed, and iii) WP generation during stop-to-operation transitions and vice versa [15]. Modeling normal operation, thus, requires data cleaning as proposed in the literature by labeling normal operation based on expert knowledge [10] or automating data cleaning by removing samples that significantly differ from an already fitted WP curve [9], or where the power generation is lower than a threshold near zero [9, 12]. The challenge of identifying inconsistent data is also present for autoregressive methods applied to WP forecasting. Such methods consider a horizon of past WP generation values to make a forecast. Most state-of-the-art autoregressive methods are based on DL architectures like the attention-based transformer architecture [16–20]. While complex relationships can be learned by such methods, their scalability is limited due to the high computational training effort.

Therefore, the present paper proposes the new method called AutoWP for day-ahead WP forecasting, which automates data pre-processing and is scalable by requiring low computational training effort while achieving competitive forecasting performance. The automated pre-processing uses a rule-based and an outlier detection-based approach, and AutoWP's model design relies on an ensemble of diverse WP curves that is fitted to the site-specific conditions of the new WP turbine. Specifically, a diverse ensemble of OEM WP curves from the Open Energy Platform (OEP) wind turbine library [21] is created, and the optimal contribution of these curves to the weighted sum of curves is determined using the least squares method.

The remainder of this paper is organized as follows. We detail the methodological concept of AutoWP in Section 2, evaluate AutoWP in Section 3, discuss the results in Section 4, and provide a conclusion and outlook in Section 5.

2 AutoWP

The core of AutoWP is to represent a new WP turbine as an optimally weighted ensemble of WP curves, similar to the method AutoPV [22, 23].[1] We first introduce the WP curve before describing the three-step ensemble method.

The WP curve $y[k] = f(v_{\text{eff}}[k])$ is a function of the power generation $y[k]$ depending on the wind speed at a reference height $v_{\text{eff}}[k]$ (commonly the hub height). Because wind speed forecasts are usually available at 10 m and 100 m, forecasting using WP curves requires a height correction[2] to account for different wind speeds at different heights due to the atmospheric boundary layer [24]. For AutoWP, we perform the height correction of the wind speed forecast based on the straightforward wind power law

$$\frac{v_{\text{a}}}{v_{\text{b}}} = \left(\frac{h_{\text{a}}}{h_{\text{b}}}\right)^{\alpha_{\text{h}}}, \tag{1}$$

with v_{a} and v_{b} being the wind speeds at heights h_{a} and h_{b} above ground level, and α_{h} depends on the terrain [25].[3] Using (1) allows to estimate the effective hub height

$$h_{\text{eff}} = 100\,\text{m} \cdot \left(\frac{\overline{v}_{\text{eff}}}{\overline{\hat{v}}_{100}}\right)^{(1/\alpha_{\text{h}})} \tag{2}$$

from the averages of the 100 m wind speed forecast $\overline{\hat{v}}_{100}$ and the hub height wind speed measurement $\overline{v}_{\text{eff}}$ over the samples in the training data sub-set. Then, wind speed forecasts $\hat{v}_{100}[k]$ can be height-corrected using

$$\hat{v}_{\text{eff}}[k] = \hat{v}_{100}[k] \cdot \left(\frac{h_{\text{eff}}}{100\,\text{m}}\right)^{\alpha_{\text{h}}}, \tag{3}$$

and used as inputs of the WP curve to make a forecast.

---

[1] AutoPV also uses an optimally weighted ensemble to represent new PV plants with differing site conditions and unknown mounting configurations (tilt and azimuth angles), as well as mixed-oriented PV plants.

[2] Except if the turbine's hub height is coincidentally 100 m. If the wind speed forecast is available at a different height, for example 80 m, the height in (2) and (3) must be adjusted accordingly.

[3] For onshore and offshore WP turbines, typical estimates for α_{h} are $1/7$, respectively $1/9$ [26].

The three-step method AutoWP entails i) creating the ensemble, ii) computing the ensemble output by the optimally weighted sum, and iii) re-scaling the ensemble output, exemplified in Figure 1, and detailed in the following.

The first step creates the WP curve ensemble using OEM curves from the OEP wind turbine library [21]. Specifically, all WP curves are resampled to ensure a uniform sampling rate of the wind speed v_{eff}, and normalized with the turbine's peak power rating $P_{\max,n}$:

$$P_n^*[v] = \frac{P_n[v]}{P_{\max,n}}. \tag{4}$$

In order to limit the ensemble size without losing diversity, we sort all normalized WP curves according to the area under the curve, retaining the first and last WP curves and eight WP curves equally spaced in between, i. e., $N_{\text{m}} = 10$.

The second step computes the ensemble WP curve based on the weighted sum of $N_{\text{m}} = 10$ WP curves

$$\hat{y}^*[k] = \sum_{n=1}^{N} \hat{w}_n \cdot \hat{y}_n^*[k], \tag{5}$$

with the weight $\hat{w}_n$, $n \in \mathbb{N}_1^N$ and the output $\hat{y}_n$ of the n-th normalized WP curve. The optimal weights are determined by minimizing the mean squared error

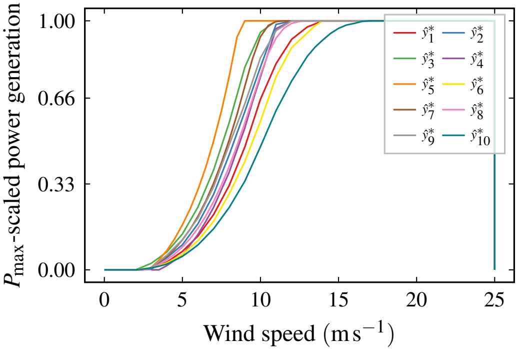
$$\min_{\hat{\boldsymbol{w}}} \frac{1}{K} \sum_{k=1}^{K} \left(\hat{y}^*[k] - y^*[k]\right)^2 \quad \text{s.t.} \quad \hat{\boldsymbol{w}} \in [0,1], \sum_{n=1}^{N} \hat{w}_n = 1, \tag{6}$$

between ensemble output $\hat{y}^*$ (5) and the target time series y^*, which is normalized analogously to (4).[4] We solve the optimization problem (6) with the least squares algorithm of the Python package SciPy [27] and normalize the weights to ensure a convex linear combination.[5]

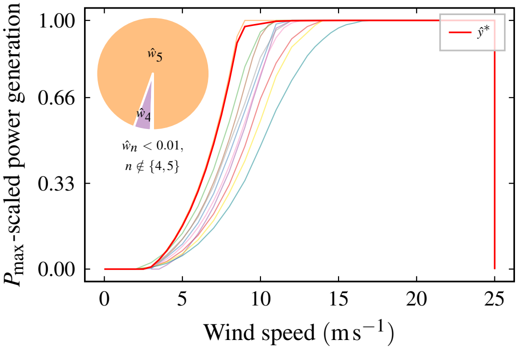
---

[4] $\hat{w}_n = 1/N$, $\forall n \in [1,N]$ can be used for initialization if no data is available (cold-start problem), to be adapted during operation as measurements become available, as shown for AutoPV [22,23].

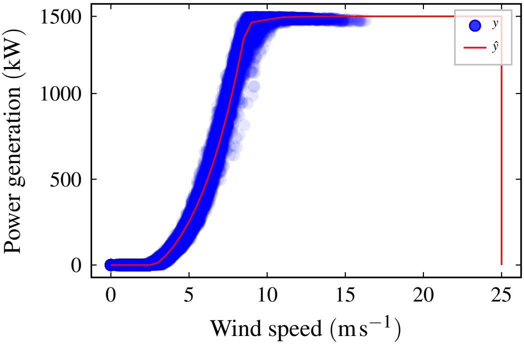
[5] Alternatively, the constraints in (6) could be relaxed, i. e., weights greater than one are accepted, and the sum of the weights does not have to be one. Consequently, the re-scaling in (7) is obsolete as the sum of weights represent the new WP turbine's peak power rating, making AutoWP also suitable for WP turbines with unknown peak power ratings.



(a) The first step creates the ensemble pool using $N_{\mathrm{m}} = 10$ normalized WP curves $\hat{y}^*$, $n \in \mathbb{N}_1^{\mathrm{m}}$ of the OEP wind turbine library [21]. The selection reduces redundancy and preserves diversity.



(b) The seconds step forms the normalized ensemble WP curve $\hat{y}^*$ as convex linear combination of the pool's curves, with the weights $\hat{w}_n$, $n \in \mathbb{N}_1^{N_{\mathrm{m}}}$ adapted to optimally fit the new WP turbine.



(c) The third step re-scales the ensemble WP curve $\hat{y}$ to the peak power rating of the new WP turbine, in this example $P_{\text{max,new}} = 1500\,\text{kWp}$. Additionally, the measurement $\hat{y}$ used to fit the ensemble weights is shown.

Figure 1: The three steps of AutoWP's automated design exemplified for the real-world WP turbine no. 2.

The third step re-scales the ensemble output (5) using the new WP turbine's peak power rating $P_{\text{max,new}}$:

$$\hat{y}[k] = \hat{y}^{*}[k] \cdot P_{\text{max,new}}. \tag{7}$$

3 Evaluation

In this section, we benchmark AutoWP against the OEM WP curve, ML-based WP curve modeling methods, and an autoregressive DL method.

Data We evaluate AutoWP on a real-world data set of the quarter-hourly energy metering (kWh) in 2019 and 2020 from two WP turbines with a peak power rating of 1500 kWp.[6] This data set is useful for the evaluation, as the WP turbines are subject to unique shutdown patterns, as shown below. The energy metering time series of both WP turbines are transformed into mean power generation time series

$$\overline{P}[k] = \frac{\Delta E[k]}{t_k}, \tag{8}$$

to increase interpretability, with the energy generation $\Delta E[k]$ (kWh) metered within the sample period t_k (h). Additionally, average wind speed measurements at hub height, as well as day-ahead weather forecasting data from the European Centre for Medium-Range Weather Forecasts (ECMWF) [28] with the origin at 00:00 are available. We split the data into a training data sub-set (2019) and a test data sub-set (2020).[7]

Both WP turbines are located in southern Germany and are subject to many shutdowns. Abnormal operating states in the data are identified according to [9] and [15]: All data points with a wind speed measurement greater than the cut-in speed and a power generation lower than the cut-in power are considered

---

[6] Dynamics below the 15-minute resolution like stop-to-operation transitions and vice versa, as well as wind gusts, cannot be considered.

[7] The forecasting method TFT additionally holds-out validation data of 20 % of the training data to terminate training if the validation loss increases (early stopping).

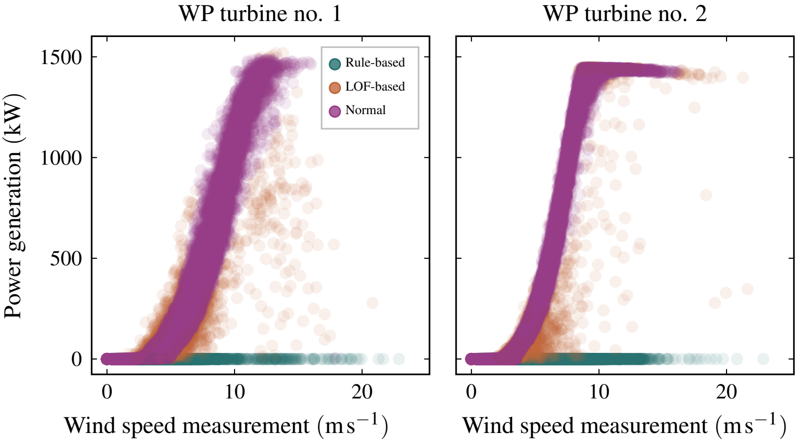


Figure 2: Data pre-processing to identify abnormal operational states. Rule-based filtering identifies turbine shutdowns and LOF-based filtering identifies turbine stop-to-operation transitions and vice versa. The samples for which the wind speed at hub height is below the cut-in speed $v_{\text{cut-out}} = 2.5\,\mathrm{m\,s^{-1}}$ amount to 16.7 %, while the cut-out speed $v_{\text{cut-out}} = 25\,\mathrm{m\,s^{-1}}$ at which the WP turbine would be shut down for safety reasons is not reached.

as shutdowns [9], and the Local Outlier Factor (LOF) algorithm is applied to identify outliers due to WP turbine stop-to-operation transitions and vice versa [15], see Figure 2. Since we aim to model the normal operating state of the WP turbine, abnormal operating states are dropped from the training data sub-set.

Benchmarks We compare the forecasting error of AutoWP against the OEM WP curve, ML-based WP curve modeling methods, and an autoregressive DL method. Regarding WP curve modeling, we consider established non-linear regression methods to directly learn the weather-dependent WP curve using the explanatory variables air temperature and atmospheric pressure, as well as wind speed and direction. Specifically, we consider the established methods eXtreme Gradient Boosting (XGB), SVR, and the MLP that work well in a variety of regression tasks. Regarding autoregressive DL-based forecasting, we consider the TFT [29]. This method combines recurrent Long Short-Term Memory (LSTM) layers with the attention-based transformer architecture. Recurrent layers allow the output from neurons to affect the subsequent input to the same neuron, which is beneficial for capturing short-term dependencies. Attention units compute importance scores for each element in the input sequence, which is useful for capturing long-term dependencies. For the day-ahead forecast with the origin at 00:00, we consider past values up to one day (96 values), and covariate time series of air temperature and atmospheric pressure, as well as wind speed and direction.

The TFT is implemented with the Python package PyTorch Forecasting [30], SVR and MLP with the Python package scikit-learn [31], and XGB with the authors' [32] Python package. For all methods, we use the default hyperparameter configuration of the respective Python package, and methods that use a stochastic training algorithm (TFT and MLP) are run five times.

Evaluation strategy In the evaluation, we assume knowing the times at which the WP turbine will be shut down, which is reasonable for day-ahead forecasts, as the turbine operator should be aware of regular shutdowns like maintenance work or bat protection. That is, we calculate assessment metrics only for samples identified as normal operation. As assessment metrics, we consider the normalized Mean Absolute Error (nMAE)

$$\mathrm{nMAE} = \frac{\sum_{k=1}^{K} |\hat{y}[k] - y[k]|}{\sum_{k=1}^{K} y[k]}, \tag{9}$$

and the normalized Root Mean Squared Error (nRMSE)

$$\mathrm{nRMSE} = \frac{\sqrt{\frac{1}{K}\sum_{k=1}^{K} (\hat{y}[k] - y[k])^2}}{\frac{1}{K}\sum_{k=1}^{K} y[k]}, \tag{10}$$

where $\hat{y}[k]$ is the forecast and $y[k]$ the realized value at time point k.

Results Table 1 shows the results on the test sub-set of the evaluation data. We observe that the OEM WP curve achieves comparably higher forecasting errors for WP turbine no. 1. Among methods based on WP curve modeling, AutoWP, MLP, and SVR achieve a similar forecasting error. While AutoWP achieves

Table 1: The forecasting errors on the test data sub-set of AutoWP compared to other methods based on WP curve modeling (OEM curve, MLP, SVR, XGB), and to the autoregressive DL method TFT.

Turbine no.	Error	AutoWP	OEM curve	MLP	SVR	XGB	TFT
1	nMAE	**0.69**±0.00	0.94±0.00	0.83±0.01	0.70±0.00	0.84±0.00	0.74±0.00
2		**0.61**±0.00	0.67±0.00	0.69±0.01	0.64±0.00	0.74±0.00	0.67±0.00
1	nRMSE	1.2±0.00	1.57±0.00	1.15±0.01	**1.11**±0.00	1.22±0.00	1.26±0.00
2		0.93±0.00	1.07±0.00	**0.92**±0.00	0.93±0.00	1.01±0.00	0.98±0.00

the lowest nMAE for both turbines, the SVR achieves the lowest nRMSE for WP turbine no. 1, and the MLP for no. 2. The autoregressive DL method TFT achieves competitive performance but requires a high computational training effort, see Table 2.[8] In contrast, the WP curve modeling approaches require substantially less training time. Since the OEM WP curve only requires height correction, its training effort is the lowest. For XGB and AutoWP, the training time is still less than one second, whereas SVR and MLP are in the region of half a minute, and the TFT already requires well over an hour for training.

Table 2: The computational training effort in seconds of AutoWP compared to other methods based on WP curve modeling (OEM curve, MLP, SVR, XGB), and to the autoregressive DL method TFT. For the OEM curve, the effort refers to the computing time for the height correction.

AutoWP	OEM curve	MLP	SVR	XGB	TFT
0.6 ±0.3	**0.001** ±0.000	19.3 ±6.2	31.6 ±2.4	0.2 ±0	6206 ±1183

Insights Figure 3 shows a visual comparison of AutoWP (WP curve modeling) and TFT (autoregressive DL) for WP turbine no. 1. Additionally, the wind speed measurement at hub height and the height-corrected wind speed forecast from ECMWF are shown. This exemplary comparison shows that the static WP curve modeling methods (no temporal context) cannot forecast intervention-related WP turbine shutdowns, while the TFT captures them with a delay. Specifically, on the day-ahead forecast no. 4, the TFT captures the shutdown in the past

---

[8] For training, we use an Intel Xeon Platinum 8368 CPU with 76 cores and an NVIDIA A100-40 GPU with 40 GB memory (the GPU is only used for the DL-based TFT), and 256 GB RAM provided by the HAICORE HPC.

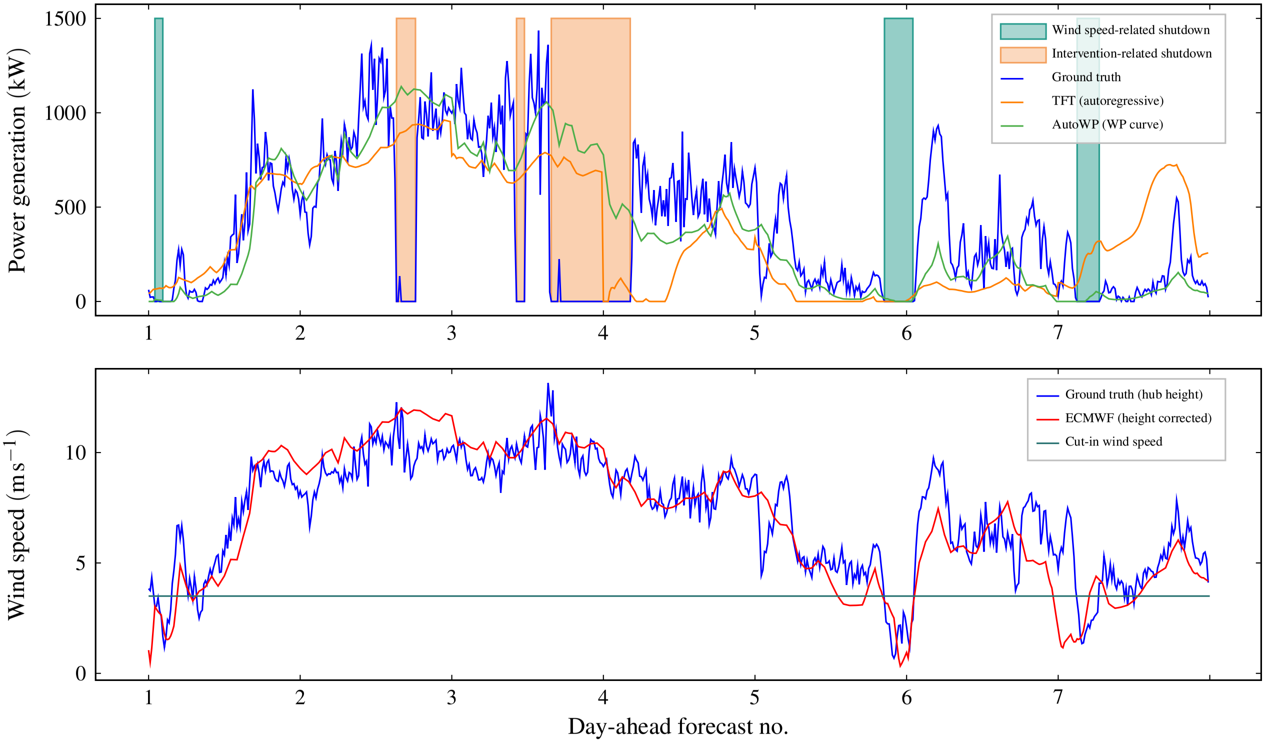


Figure 3: Exemplary comparison of seven day-ahead WP forecasts with origin at 00:00 for WP turbine no. 1 using AutoWP (WP curve modeling) TFT (autoregressive). While AutoWP is only based on wind speed forecasts (ECMWF), the TFT also considers past WP generation measurements.

horizon of the previous day. However, the shutdown in the past horizon also leads to a delayed capture of the WP turbine activation.

4 Discussion

This section discusses the benchmarking results, the limitations, and benefits of AutoWP.

Benchmarking We discuss three aspects: First, the results show that all considered methods outperform the OEM WP curve. This can be explained by the fact that the OEM WP curve does not consider site-specific conditions characterized by terrain-related air turbulence. As these conditions may be very different from those under which the OEM WP curve was determined, it is worth training the forecasting model to these conditions using measured data from the WP turbine. Second, the computational effort required to train the model varies greatly. Although the OEM curve requires the least amount of computational effort (0.001 s), the forecasting error can be reduced considerably with a small amount of effort using AutoWP (0.6 s). This is because the computational effort for solving the least squares optimization problem to determine AutoWP's optimal ensemble weights is similar to linear regression, but AutoWP can represent the non-linear progression of the curve through the ensemble of WP curves. Third, it is apparent that the methods based on WP curve modeling cannot forecast interventions into the WP generation capability. Although in the autoregressive DL method TFT, past shutdowns contribute to a certain extent to forecast future shutdowns, shutdowns are only captured with a time delay. On the one hand, this means that additional information is required if future shutdowns *must* be reliably forecasted and, on the other hand, that shutdowns in the past horizon require data imputation if future shutdowns *must not* be forecasted.

Limitations We discuss three aspects: First, although we consider two years of measurements from two WP turbines, the data set is rather small to draw generalizable conclusions. However, the results show that AutoWP can significantly reduce the forecasting error while requiring only a small amount of computational effort. While the performance of the benchmarks could potentially be improved by hyperparameter optimization, it would further increase their computational effort. Second, recurring shutdown patterns are visible in the data (see Figure 4), which can potentially be forecasted using additional information. Although the results show that shutdowns in the past horizon of the autoregressive DL method TFT contributes to the forecast, further evaluation is necessary, e. g., shutdown imputation in the past horizon, additional features for encoding the seasonality, and considering the theoretical WP generation during shutdowns. Third, the measured wind speed at hub height does not reach the cut-out speed $v_{\text{cut-out}} = 25\,\mathrm{m\,s^{-1}}$, at which the WP turbine would be shut down for safety reasons. However, as AutoWP also accounts for these shutdowns in the ensemble WP curve (see Figure 1), application to WP turbines in regions with higher wind speeds is possible.

Benefits AutoWP is beneficial since it can represent various site conditions using the ensemble of diverse WP curves. Specifically, the site conditions are

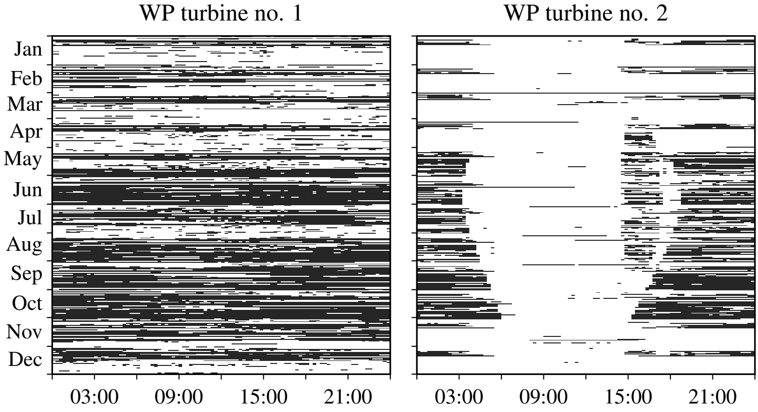


Figure 4: Identified WP turbine shutdowns with rule-based filtering. Black fields represent data points at a measured wind speed greater than the cut-in speed with a power generation lower than the cut-in power.

characterized by terrain-related air turbulence, which affects the wind speed where the WP reaches its peak power output. Although ML methods can also be used to train a site-specific WP curve, AutoWP implicitly considers physical limitations in WP generation and requires only a few data to train the model. This scalability is an essential prerequisite for deploying the model to hundreds of decentral located onshore WP turbines to serve smart grid applications like redispatch planning.

5 Conclusion and outlook

Many smart grid applications require forecasts of locally distributed WP generation. In order to handle the increasing demand for WP forecasting models, scalable methods are required that can be trained efficiently to represent the site-specific conditions characterized by the terrain-dependent air turbulence, while being robust in terms of inconsistent training data. AutoWP addresses these challenges, providing a scalable day-ahead forecasting method that includes the identification and filtering of inconsistent training data and implicitly considers physical limitations in WP generation. While achieving competitive performance against other WP curve modeling methods, the evaluation shows that such methods require knowledge about scheduled interventions into the WP generation capabilities to consider future shutdowns in the forecast. This limitation is overcome to some extent by the benchmarking method TFT, which is a state-of-the-art autoregressive DL method that considers past WP generation values, also including shutdowns. The TFT, however, captures shutdowns with a time delay and requires an enormous training effort.

Therefore, future work could evaluate WP turbines from different regions, and different shutdown handling methods for autoregressive WP forecasting methods. Furthermore, since only the TFT was evaluated, it is an open question how other DL-based autoregressive methods perform in WP forecasting in the presence of regular and irregular shutdowns.

Acknowledgements

This project is funded by the Helmholtz Association under the Program "Energy System Design", the German Research Foundation (DFG) as part of the Research Training Group 2153 "Energy Status Data: Informatics Methods for its Collection, Analysis and Exploitation" and the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI, and is supported by the HAICORE@KIT partition. Furthermore, the authors thank Stadtwerke Karlsruhe Netzservice GmbH (Karlsruhe, Germany) for the data required for this work.

References

[1] H. Xu, Y. Chang, Y. Zhao, and F. Wang. A new multi-timescale optimal scheduling model considering wind power uncertainty and demand response. *International Journal of Electrical Power & Energy Systems*, vol. 147, p. 108832, 2023. doi:10.1016/j.ijepes.2022.108832.

[2] Z. Cai, C. Bussar, P. Stöcker, L. Moraes, D. Magnor, and D. U. Sauer. Optimal Dispatch Scheduling of a Wind-battery-System in German Power Market. *Energy Procedia*, vol. 99, pp. 137–146, 2016. doi:10.1016/j.egypro.2016.10.105.

[3] F. Salm, M. Oettmeier, and P. Rönsch. The new redispatch and the impact on energy management in districts. In *Proceedings of the 2022 18th International Conference on the European Energy Market (EEM)*, pp. 1–4, 2022. doi:10.1109/EEM54602.2022.9921014.

[4] R. Meka, A. Alaeddini, and K. Bhaganagar. A robust deep learning framework for short-term wind power forecast of a full-scale wind farm using atmospheric variables. *Energy*, vol. 221, p. 119759, 2021. doi:10.1016/j.energy.2021.119759.

[5] M. Lydia, S. S. Kumar, A. I. Selvakumar, and G. E. Prem Kumar. A comprehensive review on wind turbine power curve modeling techniques.

Renewable and Sustainable Energy Reviews, vol. 30, pp. 452–460, 2014. doi:10.1016/j.rser.2013.10.030.

[6] J. Cidrás, E. Díaz-Dorado C. Carrillo, A. F. Obando Montaño, J. Cidrás, J. and E. Díaz-Dorado. Review of power curve modelling for wind turbines. *Renewable and Sustainable Energy Reviews*, vol. 21, pp. 572–581, 2013. doi:10.1016/j.rser.2013.01.012.

[7] A. Arnold, S. König, R. Mikut, and P. Bretschneider. Application of data mining methods for power forecast of wind power plants. In *Proceedings of the 9th International Workshop on Large-Scale Integration of Wind Power into Power Systems as well as on Transmission Networks for Offshore Wind Power Plants*, pp. 655–60, Quebec City, Canad, 2010. Energynautics GmbH, Langen, Germany. ACM, New York, USA.

[8] R. K. Pandit, D. Infield, and A. Kolios. Comparison of advanced non-parametric models for wind turbine power curves. *IET Renewable Power Generation*, vol. 13, no. 9, pp. 1503–1510, 2019. doi:10.1049/iet-rpg.2018.5728.

[9] Y. Wang, Q. Hu, D. Srinivasan, and Z. Wang. Wind power curve modeling and wind power forecasting with inconsistent data. *IEEE Transactions on Sustainable Energy*, vol. 10, no. 1, pp. 16–25, 2019. doi:10.1109/TSTE.2018.2820198.

[10] S. R. Moreno, L. S. Coelho, H. V. H. Ayala, and V. C. Mariani. Wind turbines anomaly detection based on power curves and ensemble learning. *IET Renewable Power Generation*, vol. 14, no. 19, pp. 4086–4093, 2020. doi:10.1049/iet-rpg.2020.0224.

[11] S. Li, D. C. Wunsch, E. A. O'Hair, and M. G. Giesselmann. Using neural networks to estimate wind turbine power generation. *IEEE Transactions on Energy Conversion*, vol. 16, no. 3, pp. 276–282, 2001. doi:10.1109/60.937208.

[12] M. Schlechtingen, I. F. Santos, and S. Achiche. Using data-mining approaches for wind turbine power curve monitoring: A comparative

study. *IEEE Transactions on Sustainable Energy*, vol. 4, no. 3, pp. 671–679, 2013. doi:10.1109/TSTE.2013.2241797.

[13] U. Singh, M. Rizwan, M. Alaraj, and I. Alsaidan. A machine learning-based gradient boosting regression approach for wind power production forecasting: A step towards smart grid environments. *Energies*, vol. 14, no. 16, article 5196, 2021. doi:10.3390/en14165196.

[14] M. Á. Rodríguez-López, E. Cerdá, and P. del Rio. Modeling wind-turbine power curves: Effects of environmental temperature on wind energy generation. *Energies*, vol. 13, no. 18, article 4941, 2020. doi:10.3390/en13184941.

[15] R. Morrison, X. Liu, and Z. Lin. Anomaly detection in wind turbine SCADA data for power curve cleaning. *Renewable Energy*, vol. 184, pp. 473–486, 2022. doi:10.1016/j.renene.2021.11.118.

[16] S. Sun, Y. Liu, Q. Li, T. Wang, and F. Chu. Short-term multi-step wind power forecasting based on spatio-temporal correlations and transformer neural networks. *Energy Conversion and Management*, vol. 283, p. 116916, 2023. doi:10.1016/j.enconman.2023.116916.

[17] L. van Heerden, C. van Staden, and H. J. Vermeulen. Temporal Fusion Transformer for day-ahead wind power forecasting in the South African context. In *Proceedings of the 2023 IEEE International Conference on Environment and Electrical Engineering and 2023 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, pp. 1–5, 2023. doi:10.1109/EEEIC/ICPSEurope57605.2023.10194737.

[18] Y. Hu, H. Liu, S. Wu, Y. Zhao, Z. Wang, and X. Liu. Temporal collaborative attention for wind power forecasting. *Applied Energy*, vol. 357, p. 122502, 2024. doi:10.1016/j.apenergy.2023.122502.

[19] S. Xu, Y. Wang, X. Xu, G. Shi, Y. Zheng, H. Huang, and C. Hong. A multi-step wind power group forecasting seq2seq architecture with spatial-temporal feature fusion and numerical weather prediction correction. *Energy*, vol. 291, p. 130352, 2024. doi:10.1016/j.energy.2024.130352.

[20] S. Mo, H. Wang, B. Li, Z. Xue, S. Fan, and X. Liu. Powerformer: A temporal-based transformer model for wind power forecasting. *Energy Reports*, vol. 11, pp. 736–744, 2024. doi:10.1016/j.egyr.2023.12.030.

[21] M. Petersen, J. Huber, and C. Hofmann. Wind turbine library. Further contributors: Ludee, jh-RLI, 2019. `https://openenergyplatform.org/dataedit/view/supply/wind_turbine_library`,

[22] S. Meisenbacher, B. Heidrich, T. Martin, R. Mikut, and V. Hagenmeyer. AutoPV: Automated photovoltaic forecasts with limited information using an ensemble of pre-trained models. In *Proceedings of the 14th ACM International Conference on Future Energy Systems (e-Energy '23)*, pp. 386–414, Orlando, USA, 2023. ACM, New York, USA. doi:10.1145/3575813.3597348.

[23] S. Meisenbacher, T. Martin, B. Heidrich, R. Mikut, and V. Hagenmeyer. Automating day-ahead forecasting of photovoltaic power generation: Model design, monitoring, and adaption. In *ETG Congress 2023*, pp. 1–8, 2023.

[24] D. Etling. Theoretische Meteorologie: Eine Einführung. Chapter 21: Die atmosphärische Grenzschicht, pp. 297–340. Springer-Verlag, Berlin, Heidelberg, Germany, 2004. doi:10.1007/978-3-540-75979-9_21.

[25] G. M. Masters. Wind power systems. Chapter 6 in *Renewable and efficient electric power systems*, pp. 307–383. John Wiley & Sons, Ltd, Chichester, UK, 2004. doi:10.1002/0471668826.ch6.

[26] S. A. Hsu, E. A. Meindl, and D. B. Gilhousen. Determining the power-law wind-profile exponent under near-neutral stability conditions at sea. *Journal of Applied Meteorology and Climatology*, 33(6):757–765, American Meteorological Society, Boston, USA, 1994. doi:10.1175/1520-0450(1994)033<0757:DTPLWP>2.0.CO;2.

[27] P. Virtanen et al.. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17:261–272, 2020. doi:10.1038/s41592-019-0686-2.

[28] European Centre for Medium-Range Weather Forecasts (ECMWF). Meteorological Archival and Retrieval System (MARS). `https://confluence.ecmwf.int/display/CEMS/MARS`. Accessed: 2024-06-03, 2023.

[29] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021. doi:10.1016/j.ijforecast.2021.03.012.

[30] J. Beitner. PyTorch Forecasting. `https://towardsdatascience.com/introducing-pytorch-forecasting-64de99b9ef46`. Accessed: 2024-06-03, 2020.

[31] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(85):2825—2830, 2011.

[32] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '23)*, pp. 785–794, San Francisco, USA, 2016. ACM, New York, USA. doi:10.1145/2939672.2939785.

Datengetriebene Modellierung und Merkmalsselektion für die automatische Kalibrierung eines Sensorsystems für die mikromagnetische Materialcharakterisierung

Felix Wittich[1], Andreas Kroll[1]

[1] Fachgebiet Mess- und Regelungstechnik
Universität Kassel
Mönchebergstraße 7, 34125 Kassel
E-Mail: {felix.wittich, andreas.kroll@mrt.uni-kassel.de}

Kurzfassung

Der industrielle Echtzeiteinsatz zerstörungsfreier Prüfverfahren bietet ein großes Potenzial für die Analyse oberflächennaher Werkstückeigenschaften. Neben einer möglichen Zeit- und Kostenreduktion durch die direkte Integration von Prüfverfahren in den Bearbeitungsprozess, die bisher nur offline durchführbar war, ermöglicht dies eine Überwachung und gezielte Regelung von relevanten Werkstückeigenschaften im laufenden Prozess. Eine zentrale Herausforderung beim Einsatz mikromagnetischer Materialcharakterisierungsverfahren (MMV) [1] ist dabei deren Kalibriermodellerstellung. Die MMV können beispielsweise bei der Prozessdatenerfassung für die Prozessparameteroptimierung in der Fertigung eingesetzt werden [2], ohne dabei aufwendige ex-process Referenzmessungen durchführen zu müssen. In diesem Beitrag wird ein Ansatz für die datengetriebene Kalibriermodellerstellung für die mikromagnetische (MM) Messung vorgestellt und anhand eines Hartdrehdatensatzes untersucht.

1 Einführung

Die mikromagnetische Materialcharakterisierung ermöglicht die zerstörungsfreie in-process Messung mechanischer Eigenschaften ferromagnetischer Werkstücke. Das Potential liegt dabei in der vielfältigen Einsetzbarkeit in Bezug auf die Zielgrößen wie u. a. Eigenspannung, Härte, Härtetiefe, Streckgrenze, sowie der Breite der potentiellen Anwendungsfelder in verschiedenen Fertigungsprozessen (Urformen, Umformen, Fügen, Trennen). Der Randschichtzustand eines Bauteils bestimmt dessen Funktionalität, Belastbarkeit und Lebensdauer. Eine gezielte Einstellung dieser Eigenschaften erfordert daher eine in-process Erfassung des Randschichtzustands. Wichtige Zielgrößen wie die Eigenspannung werden bisher sehr zeitaufwendig und / oder zerstörend röntgenografisch oder mittels der Bohrlochmethode erfasst. Auch die Oberflächenhärte wird konventionell über die Eindringtiefe eines Prüfkörpers quantifiziert und die Streckgrenze wird zerstörend mittels Zugversuch bestimmt. Die vorgenannten Prüfverfahren müssen ex-process durchgeführt werden und eignen sich daher nicht für ein prozessintegriertes Steuerungs- oder Regelungskonzept. Hier bietet die mikromagnetische zerstörungsfreie Messung das Potential, aufwendige Laboruntersuchungen zeit- und kosteneffizient zu ersetzen und die Fertigungsprozesse zu regeln.

Um auf Basis von MMV die Randschichteigenschaften zu quantifizieren, muss der funktionale Zusammenhang zwischen den mikromagnetischen Messgrößen und den resultierenden Eigenschaften bekannt sein. Bei der MMV werden verschiedene Messmethoden genutzt: die Oberwellenanalyse der Magnetfeldstärke, das Barkhausenrauschen und die Analyse des Mehrfrequenzwirbelstroms sowie der Überlagerungspermeabilität. Entsprechend besteht die Herausforderung darin, die gemessenen Zeitreihen auf skalare Zielgrößen abzubilden. Eine Modellierung dieser Zusammenhänge basierend auf physikalischen Ansätzen ist aufgrund der komplexen, nicht vollständig verstandenen Zusammenhänge nicht zielführend [3]. Vielmehr hat sich ein datengetriebener Ansatz etabliert, basierend auf einfachen Regressionsansätzen mit schrittweiser Merkmalsselektion, mit dem aber teilweise keine guten Ergebnisse erzielt werden können [4]. Trotz des großen Potenzials zur Zeit- und Kostenreduzierung steht dem breiten

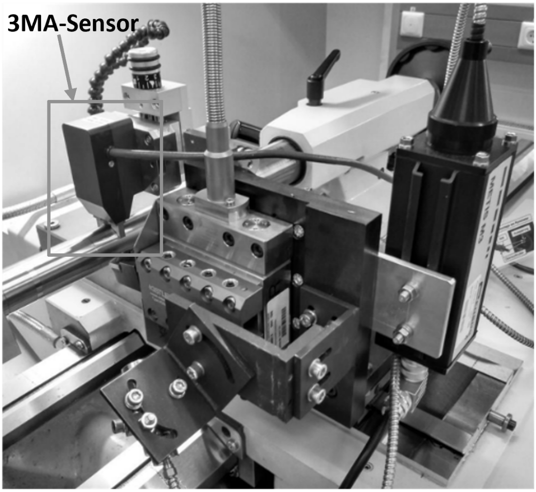


Bild 1: Versuchsaufbau mit 3MA-Sensor an der Hartdrehmaschine

industriellen Einsatz die erforderliche komplexe Kalibriermodellerstellung entgegen [5].

Verschiedene Arbeiten befassen sich mit der datengetriebenen Kalibrierung von MMV-Systemen. In [6] wird eine klassifikationsbasierte und eine kontinuierliche Quantifizierung mittels multipler linearer Regression betrachtet. Auf der in [6] durchgeführten regressionsbasierten Zugspannungsquantifizierung aufbauend, wird in [7] eine Regression mittels Multi Layer Perceptron (MLP), Support Vector Machine (SVM) und multipler linearer Regression verglichen, wobei allerdings lediglich lineare Aktivierungsfunktionen und Kernelfunktionen betrachtet werden und keine Merkmalsselektion stattfand. In [8] wird eine geeignete Gewichtung der betrachteten Merkmale zur robusten kNN-basierten Klassifikation mittels GA untersucht. In [9] wurde ein MLP mit logistischen Aktivierungsfunktionen als nichtlinearer Ansatz mit multipler linearer Regression für die Zielgröße Einsatzhärtungstiefe ohne den Einsatz einer Merkmalsselektion verglichen. Die vorgestellten Arbeiten zeigen, dass einzelne Ansätze für spezifische Anwendungen gute Ergebnisse liefern können. Eigene Untersuchungen [4] haben aber gezeigt, dass die Standardverfahren für das

Beispiel der oberflächennahen Eigenspannungsmessungen keine zuverlässigen Ergebnisse liefern. Daher soll in diesem Beitrag ein nichtlinearer Regressionsansatz mit Gaußprozessregression und Merkmalsselektion untersucht werden, der eine Kalibriermodellerstellung für die mikromagnetischen Messung des Eigenspannungszustandes ermöglicht. Die verwendeten Daten stammen dabei aus einem industriellen Hartdrehprozess. Der Versuchsaufbau hierzu ist in Abbildung 1 zu sehen.

2 Methode

In diesem Beitrag werden empirische Modellierungsansätze betrachtet, die es ermöglichen, nichtlineare Systeme auf Basis kleiner Datensätze zu modellieren. Es soll ein funktionaler Zusammenhang zwischen n Eingangsgrößen $\boldsymbol{x} \in \mathbb{R}^n$ und den Ausgangsgrößen $y \in \mathbb{R}$ hergestellt werden

$$y(l) = f(\boldsymbol{x}(l)) + e(l), l = 1, \ldots, N, \tag{1}$$

mit dem Datensatz $Z^N = \{\boldsymbol{x}(l), y(l)\}_{l=1}^N$ mit N Elementen. Die Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ist die nichtlineare Funktion, die das Systemverhalten modellieren soll. Der Term $e(l)$ berücksichtigt Messfehler und andere Störungen, sowie Modelldefizite. Die Umsetzung der Methoden erfolgte in MATLAB 2023b.

Aufgrund der hohen Anzahl an Eingangsgrößen wird ein einfaches, auf ANOVA basierendes Verfahren zur Vorselektion der Merkmale angewendet. Hierzu wird die kontinuierliche Ausgangsgröße wird in 10 gleichgroße Gruppen eingeteilt. Für jedes Merkmal wird dann eine ANOVA durchgeführt, bzw. die F-Statistik berechnet. Jeder F-Test prüft die Hypothese, dass die Gruppen der Ausgangsgröße, die nach den Werten der Prädiktorvariablen gruppiert sind, aus Populationen mit gleichem Mittelwert stammen, gegen die alternative Hypothese, dass die Populationsmittelwerte nicht alle gleich sind. Ein kleiner p-Wert der Teststatistik deutet darauf hin, dass der entsprechende Prädiktor wichtig ist. Zur Bewertung wird ein Predictor-Importance-Score in der Form

$$-\log(p) \tag{2}$$

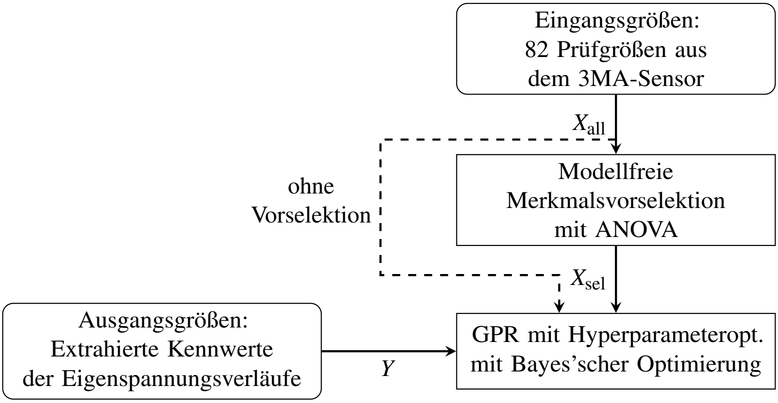


Bild 2: Vorgehen bei der Modellierung

verwendet.

Als Modellansatz wird die Gaußprozessregression (GPR) als nichtparametrische Methode verwendet. Für eine umfangreiche Beschreibung der GPR wird auf [10] verwiesen. Für die Kalibriermodellerstellung wurde GPR ausgewählt, da hiermit sehr gute Ergebnisse bei vergleichbaren Modellierungsproblemen mit kleinen Datensätzen erzielt werden konnten [2, 11]. In eigenen Versuchen mit weiteren nichtlinearen Modellansätzen wie MLPs oder SVMs konnten keine besseren Ergebnisse als mit GPR erreicht werden. Die Idee hinter einem Gauß-Prozess ist es, eine Wahrscheinlichkeitsverteilung über Funktionen zu definieren. Für die Regression wird die A-priori-Verteilung der zu erlernenden Funktion definiert als:

$$\hat{y}(l) = \mathrm{GP}(m(\boldsymbol{x}(l)), \kappa(\boldsymbol{x}(l), \boldsymbol{x}'(l))) \tag{3}$$

wobei $m(\boldsymbol{x})$ die Basis- bzw. Mittelwertfunktion und $\kappa(\boldsymbol{x}, \boldsymbol{x}')$ die Kovarianzfunktion für zwei Punkte $\boldsymbol{x}$ und $\boldsymbol{x}'$ im Eingangsraum ist (Index l im Folgenden weggelassen):

$$m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})] \tag{4}$$

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))^{\top}] \tag{5}$$

Tabelle 1: Entwurfsentscheidungen des Gaußprozessmodells bei BO

Hyperparameter	Varianten
Basisfunktion	{keine; konstant; linear; quadratisch}
Kernelfunktion	{exponentiell; rational-quadratisch; exponentiell-quadratisch; Matern32; Matern52; exponentiell mit ARD; rational-quadratisch mit ARD exponentiell-quadratisch mit ARD; Matern32 mit ARD; Matern52 mit ARD }
Standardisierung	{true; false}

mit einen positiv-definiten Kernel $\kappa(\boldsymbol{x},\boldsymbol{x}')$. Der Gauß-Prozess definiert eine gemeinsame Gauß-Verteilung für eine endliche Menge von Punkten:

$$p(y|\boldsymbol{X}) = \mathcal{N}(y|\boldsymbol{m},\boldsymbol{K}) \tag{6}$$

mit der Designmatrix (zusammengesetzt aus den Eingangsdaten) $\boldsymbol{X} \in \mathbb{R}^{N\times n}$, der Kovarianzmatrix $\boldsymbol{K} \in \mathbb{R}^{N\times N}, K_{i,j} = \kappa(\boldsymbol{x}(i),\boldsymbol{x}(j))$, und dem Mittelwertvektor $\boldsymbol{m} = [m(\boldsymbol{x}(1))\dots m(\boldsymbol{x}(N))]^T \in \mathbb{R}^N$. Der Kernel spielt eine Schlüsselrolle in Gauß-Prozessen, da er die Abhängigkeiten zwischen den Datenpunkten beschreibt. Dies beeinflusst das Verhalten des Modells maßgeblich.

Um die Ausgabe y_* für einen neuen Eingangswert $\boldsymbol{x}_*$ vorherzusagen, wird die bedingte A-posteriori-Verteilung berechnet, indem die multivariate Gauß-Verteilung konditioniert wird (Mittelwert von 0 und kein Rauschen werden zur Vereinfachung angenommen):

$$p(y_*|\boldsymbol{x}_*,\boldsymbol{X},\boldsymbol{y}) = \mathcal{N}(y_*|m_*,\kappa_*) \tag{7}$$

mit dem neuen Eingangswert $\boldsymbol{x}_* \in \mathbb{R}^n$ und dem Vektor der Ausgangsgrößen $\boldsymbol{y} \in \mathbb{R}^N$.

Während die Kernel-Parameter und die Varianz des Rauschens beim Modelltraining durch Maximierung der Log-Marginal-Likelihood-Funktion bestimmt werden können, ist dies für weitere Hyperparameter nicht möglich. Daher wird hier die Bayes'sche-Optimierung (BO) eingesetzt, um die Basisfunktion und die Kernelfunktion auszuwählen. Zudem werden die beiden Fälle standardisierter und nicht-standardisierter Daten betrachtet. Die Kernelparameter und die Rauschvarianz werden bei der BO mitoptimiert. Als Gütemaß wird dabei der kreuzvalidierte MSE herangezogen. In Tabelle 1 sind die Hyperparameter und deren Variationen dargestellt, die während der BO optimiert werden. Die BO wurde mit der Matlab-Funktion „bayesopt" umgesetzt, die als Surrogatfunktion für die zu optimierende Zielfunktion ebenfalls ein GPR-Modell verwendet. Das zusammengefasste Vorgehen bei der Modellierung ist in Abbildung 2 dargestellt.

Die mit dem 3MA-Sensor bereitgestellte Toolbox verwendet eine multiple lineare Regression (MLR) folgender Form ohne Interaktionsterme

$$f(\boldsymbol{x}) = \sum_{k=1}^{n} a_k x_k^{b_k}, a_k \in \mathbb{R}, b_k \in \{0;0,5;12\} \tag{8}$$

mit einer schrittweisen Vorwärtsselektion und einer quadratischen Gütefunktion. Die Signifikanz eines hinzugefügten Modellterms wird mittels eines F-Tests bewertet. Dieser bewertet schrittweise die Güte von zwei Modellen, mit verschiedenen Teilmengen von Regressoren, ausgehend von einem konstanten Modell und folgt einer 'greedy' Suchstrategie folgt.

3 Daten

Die Daten stammen aus einem industriellen Hartdrehprozess mit vergütetem Stahl 51CrV4. Die Proben wurden bei unterschiedlichen Temperaturen gehärtet und angelassen, sodass sich drei verschiedene Ausgangshärtegrade ergaben: 400 HV, 500 HV und 600 HV. Durch die anschließende Hartdrehbearbeitung wurden die Prozessparameter nach einem vollfaktoriellen Versuchsplan variiert, wodurch verschiedene Oberflächenzustände der Werkstücke resultierten. Je Härtecharge wurden 27 Proben mit variierten Prozessparameterkombinationen erzeugt. Insgesamt sind also 81 Proben vorhanden. Die MM-Messungen wurden

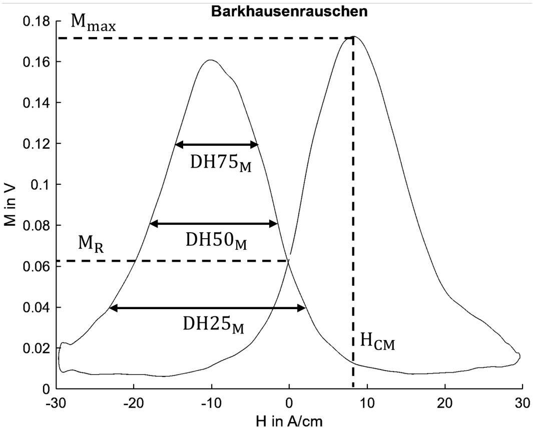


Bild 3: Beispielhafte Darstellung der sechs ermittelten Prüfgrößen aus den Messungen des Barkhausenrauschens

mit einem 3MA-II-Gerät des Fraunhofer IZFP durchgeführt. Der Versuchsaufbau mit dem 3MA-Sensor an der der Drehmaschine ist ins Abbildung 1 zu sehen.

Die Referenzmessungen der Eigenspannungen wurden mittels Röntgendiffraktometrie (XRD) ermittelt. Das 3MA-Gerät liefert für eine Messung 41 Prüfgrößen, die aus den gemessenen Zeitreihen abgeleitet werden. In Abbildung 3 sind beispielhaft 6 Prüfgrößen dargestellt, die aus einer Zeitreihe der Messung des Barkhausenrauschens abgeleitet werden. Dabei bezeichnet M_{max} die maximale Rauschamplitude, M_R die Rauschamplitude im Remanenzpunkt, H_{CM} die Feldstärke im Rauschmaximum und DHx_M die Kurvenaufweitung bei x % der Kurvenhöhe. Für eine genauere Beschreibung der Prüfgrößen, auch der weiteren Messverfahren, wird auf [6] verwiesen. Die Messungen wurden mit zwei Magnetisierungsamplituden durchgeführt, wodurch $2 \times 41 = 82$ Prüfgrößen resultieren, die als Merkmale, bzw. potentielle Eingangsgrößen für das Kalibriermodell dienen. In Abbildung 4 ist die Korrelationsmatrix für den paarweisen Vergleich der Prüfgrößen dargestellt. Dabei handelt es sich bei den Größen 1-41 und 42-81 jeweils um die Prüfgrößen für die beiden Magnetisierungsamplituden. Es ist erkennbar, dass teilweise starke Korrelationen zwischen den Prüfgrößen auftreten, was eine Merkmalsselektion motiviert. So sind Blöcke mit stark

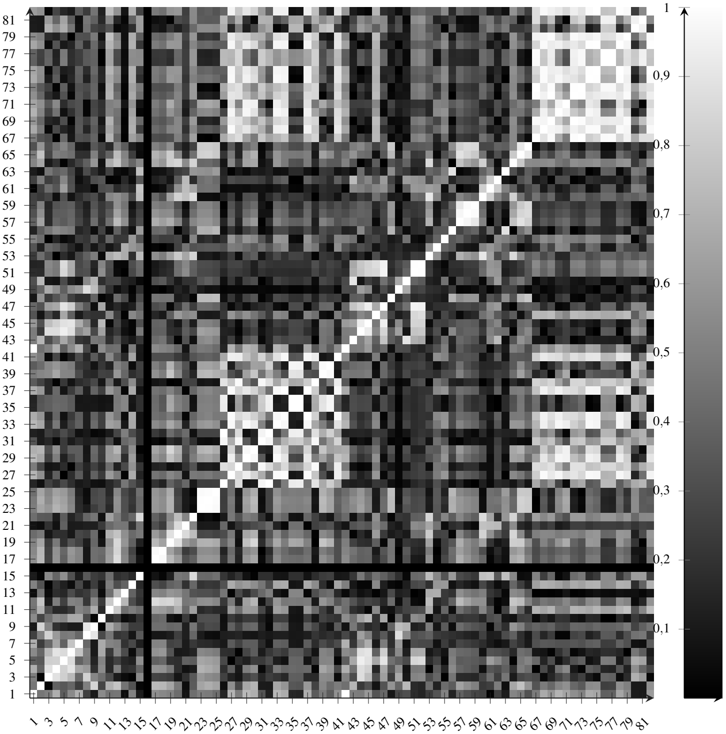


Bild 4: Korrelationsmatrixplot der Eingangsgrößen (Pearson-Korrelations-Koeffizient)

korrelierten Größen erkennbar. Dabei handelt es sich um die Kenngrößen aus den Wirbelstrommessungen, die offenbar teilweise redundante Informationen liefern.

Aus den mit XRD gemessenen Eigenspannungsverläufen in den ersten 200 µm der Werkstücke werden charakteristische Kennwerte extrahiert, die dann als Ausgangsgröße für die Kalibriermodelle eingesetzt werden. Aus Werkstoff- und Produktionstechnischer Sicht sind hierbei die Oberflächeneigenspannung sowie die maximale Druckeigenspannung von Interesse, da diese erheblichen Einfluss

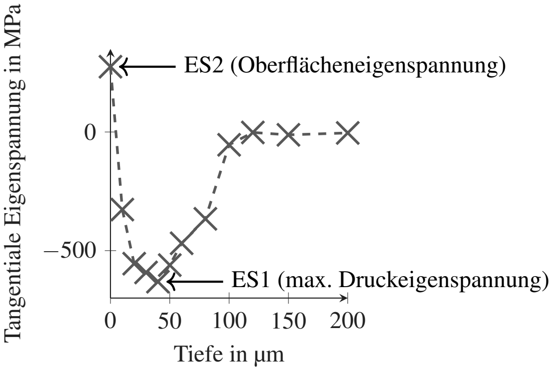


Bild 5: Beispielhafter Eigenspannungstiefenverlauf mit den charakteristischen Kennwerten

auf die Bauteileigenschaften wie Ermüdungsfestigkeit und Bruchzähigkeit haben [12]. In Abbildung 5 sind die Kennwerte in einem beispielhaften mit XRD gemessenen Eigenspannungstiefenverlauf dargestellt. ES1 ist dabei die Oberflächeneigenspannung und ES2 die maximale Druckeigenspannung.

4 Ergebnisse

Für die Bewertung der Modellgüte wird das für Regressionsmodelle übliche empirische Bestimmtheitsmaß R^2

$$R^2 = 1 - \frac{\sum_{l=1}^{N} (y(l) - \hat{y}(l))^2}{\sum_{l=1}^{N} (y(l) - \bar{y})^2} \tag{9}$$

mit dem Mittelwert $\bar{y}$ verwendet. Um die Generalisierungsfähigkeit der Modelle zu bewerten, wird ein k-fach kreuzvalidiertes (CV) Bestimmtheitsmaß R^2_{CV} verwendet. Dabei wird $k = 30$ gewählt, um bei dem kleinen Datensatz in jedem der k Folds relativ große Trainingsdatensätze zu erhalten. In Abbildung 6 sind die Predictor-Importance-Scores (Gleichung (2)) der 82 gerankten Merkmale dargestellt. Die Anzahl der vorselektierten Merkmale wird mit 25 festgelegt, da hier ein Knick im Ranking mit einem Abfall des Scores zu erkennen ist. Die Ergebnisse für GPR und MLR sind in Tabelle 2 für die Zielgrößen ES2

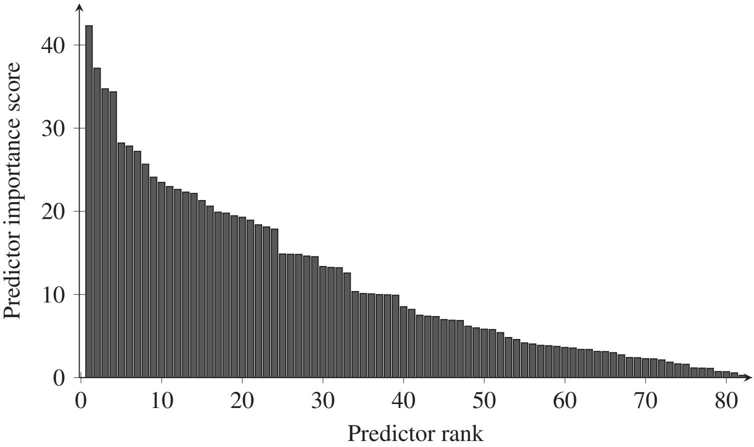


Bild 6: Ranking der Merkmale nach Predictor-Importance-Score für Zielgröße ES2

und ES1 dargestellt. Die GPR liefert deutlich bessere Ergebnisse als die MLR. Die Variante mit der Vorselektion liefert etwas bessere Ergebnisse und zeichnet sich daneben durch einen geringeren Rechenaufwand beim Training des GPR aus. Die mit BO ermittelten Hyperparameter der GPR-Modelle sind in Tabelle 3 dargestellt. Während die Standardisierung der Daten für beide Zielgrößen angewendet wird, wurden unterschiedliche optimale Basis- und Kernelfunktionen bestimmt. In Abbildung 7a und 7b ist ein Vergleich der prädizierten zu den wahren Werten für die Validierungsdaten der GPR mit Vorselektion sowie der MLR dargestellt. Bei MLR ist auffällig, dass die Streuung insbesondere bei den Werten um 0 sehr groß ist. Diese Ausreißer werden durch das GPR besser abgebildet. In Abbildung 7c und 7d sind die Fehler e in Abhängigkeit der prädizierten Werte dargestellt. Es ist erkennbar, das das GPR im Vergleich zum MLR nur wenige Fehler $e > |200|$ MPa aufweist, also zu deutlich weniger großen Abweichungen in den Prädiktionen führt.

Um die Ergebnisse mit dem GPR-Model mit Vorselektion (welches im obigen Vergleich die besten Ergebnisse erzielte) direkt mit den Ergebnissen mit dem Standardverfahren für die Kalibrierung des 3MA-Sensors zu vergleichen, wird das selbe Szenario wie in [4] verwendet. Hier wird nur die 500 HV-Charge. Da ein ursprünglich ein vollfaktorieller Experimententwurf mit 3 Faktoren

Tabelle 2: Vergleich der Modellgüten mit Kreuzvalidierung für die Zielgrößen ES2 (maximale Druckeigenspannung) und ES1 (Oberflächeneigenspannung)

Ausgangsgröße	Modell	R^2_{CV}
ES2	GPR ohne Vorselektion	0,9035
	GPR mit Vorselektion	0,9169
	MLR mit SWR wie in [4]	0,692
ES1	GPR ohne Vorselektion	0,8836
	GPR mit Vorselektion	0,8901
	MLR mit SWR wie in [4]	0,6452

Tabelle 3: Mit BO ermittelte Hyperparameter der GPR-Modelle für die Zielgrößen ES1 und ES2.

	ES1	ES2
Basisfunktion	konstant	linear
Kernelfunktion	rational-quadr. m. ARD	exponentiell-quadr.
Standardisierung	true	true

und 3 Faktorstufen verwendet wurde, ergibt sich also ein Datensatz mit $N = 27$. Es werden 5 Hold-Out(HO)-Daten für die Validierung eingesetzt. Die Ergebnisse sind in Tabelle 4 für ES2 und ES1 dargestellt. Auch hier sind die Ergebnisse mit dem GPR deutlich besser. Allerdings ist die Validierung mit 5 Datenpunkten bei 22 Trainingsdaten nur bedingt aussagekräftig, im Vergleich zur Kreuzvalidierung. Zudem ist der Trainingsdatensatz, der nur die 500-HV-Charge beinhaltet, sehr klein.

5 Zusammenfassung und Ausblick

In diesem Beitrag wurde ein Ansatz für die datengetriebene Modellierung für die Erstellung von Kalibriermodellen vorgestellt. Es konnte für ein mikromagnetisches Sensorsystem gezeigt werden, dass der Ansatz zur nichtparametrischen Modellierung gut geeignet ist und eine Modellgüte erreicht, die einen Einsatz

Tabelle 4: Vergleich der Modellgüten mit Hold-Out-Testdaten, wie in [4] für die Zielgrößen ES2 (maximale Druckeigenspannung) und ES1 (Oberflächeneigenspannung)

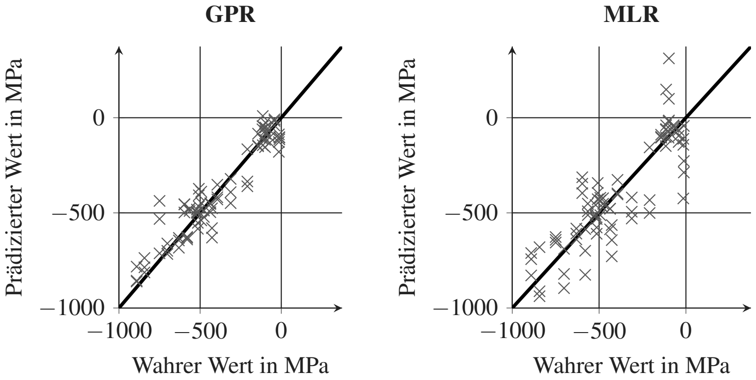
Ausgangsgröße	Modell	R^2_{HO}
ES2	GPR mit Vorselektion	0,8625
	MLR mit SWR wie in [4]	0,6317
ES1	GPR mit Vorselektion	0,8731
	MLR mit SWR wie in [4]	0,6474

in der Prozessüberwachung oder für eine Prozesssteuerung bzw. -regelung ermöglicht.

Dabei wurde ein einfaches Verfahren zur Vorselektion der Merkmale verwendet. In der Zukunft sollen hier weitere Wrapper-, Filter-, und Embedded-Verfahren untersucht werden. Darauf aufbauend soll das Verfahren um eine ensemblebasierte Merkmalsextraktion und -selektion erweitert werden. Dabei sollen die Rohdaten, also die gemessenen Zeitreihen des mikromagnetischen Sensors, verwendet werden. Damit soll es möglich sein, mit einem automatisierten Ansatz die Kalibriermodellerstellung für eine große Bandbreite an Werkstoffen und Zielgrößen zu ermöglichen. Neue Experimente und Messungen für diese Untersuchungen werden momentan durchgeführt. Dabei wird auch die Oberflächenhärte erfasst, um die Übertragbarkeit auf andere Zielgröße zu untersuchen.

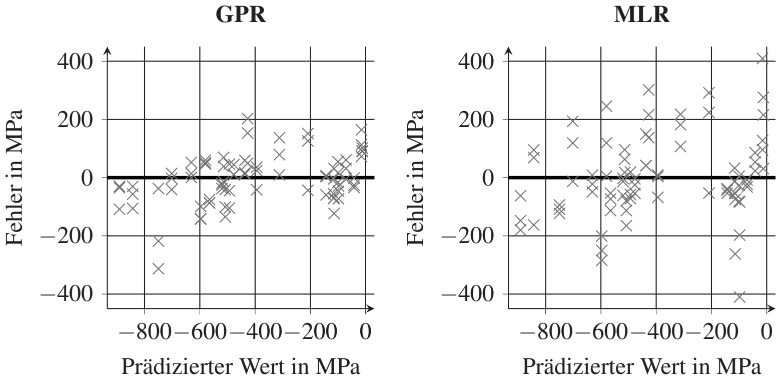
6 Danksagung

Gefördert durch die Deutsche Forschungsgemeinschaft (DFG) - Projektnummer 532921704. Die Autoren danken dem Fachgebiet Metallische Werkstoffe der Universität Kassel für die Durchführung der Hartdrehversuche sowie die Durchführung der Referenz- sowie mikromagnetischen Messungen.



(a) Wahre Werte vs. prädizierte Werte für GPR mit Vorselektion (Validierungsdaten)

(b) Wahre Werte vs. prädizierte Werte für MLR (Validierungsdaten)



(c) Prädizierte Werte vs. Fehler für GPR mit Vorselektion (Validierungsdaten)

(d) Prädizierte Werte vs. Fehler für MLR (Validierungsdaten)

Bild 7: Vergleich von MLR und GPR bezüglich der Prädiktionen (a), (b) und Fehler (c), (d) für die Validierungsdaten der Modelle für ES2

Literatur

[1] G. Dobmann, W. A. Theiner, and R. Becker, *Progress in the Micromagnetic Multiparameter Microstructure and Stress Analysis (3MA)*, p. 516–523. Springer Berlin Heidelberg, 1989.

[2] F. Wittich, T. Wegener, A. Liehr, W. Zinn, T. Niendorf, and A. Kroll, “Data-driven prediction of the surface layer state in hard-turning for optimization of component quality,” *Production Engineering*, vol. 18, p. 381–392, Jan. 2024.

[3] B. Wolter, Y. Gabi, and C. Conrad, “Nondestructive testing with 3MA—an overview of principles and applications,” *Applied Sciences*, vol. 9, no. 6, 2019.

[4] T. Wegener, A. Liehr, A. Bolender, S. Degener, F. Wittich, A. Kroll, and T. Niendorf, “Calibration and validation of micromagnetic data for non-destructive analysis of near-surface properties after hard turning,” *HTM Journal of Heat Treatment and Materials*, vol. 77, no. 2, pp. 156–172, 2022.

[5] N. Baak, R. Hajavifard, L. Lücker, J. Rozo Vasquez, S. Strodick, M. Teschke, and F. Walther, “Micromagnetic approaches for microstructure analysis and capability assessment,” *Materials Characterization*, vol. 178, Aug. 2021.

[6] S. Youssef, C. Zimmer, K. Szielasko, and A. Schütze, “Bewertung subjektiver und automatisierter Merkmalsextraktion periodischer Zeitsignale am Beispiel des 3MA-X8-Verfahrens,” *tm - Technisches Messen*, vol. 86, pp. 267–277, Jan. 2019.

[7] S. Youssef, C. Zimmer, K. Szielasko, Z. K. Suri, and A. Schütze, “Vergleich subjektiver und automatisierter Merkmalsextraktion sowie Einsatz maschineller Lernalgorithmen zur mikromagnetischen Materialcharakterisierung,” in *Tagungsband 20. GMA/ITG-Fachtagung Sensoren und Messsysteme*, pp. 347–354, 2019.

[8] K. Szielasko, B. Wolter, R. Tschuncky, and S. Youssef, “Micromagnetic materials characterization using machine learning: Progress in nondestructive prediction of mechanical properties of steel and iron,” *tm - Technisches Messen*, vol. 87, pp. 428–437, Oct. 2019.

[9] R. Jedamski and J. Epp, “Non-destructive micromagnetic determination of hardness and case hardening depth using linear regression analysis and artificial neural networks,” *Metals*, vol. 11, Dec. 2020.

[10] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[11] F. Wittich, L. Kistner, A. Kroll, C. Schott, and T. Niendorf, “On data-driven nonlinear uncertainty modeling: Methods and application for control-oriented surface condition prediction in hard turning,” *tm - Technisches Messen*, vol. 87, pp. 732–741, oct 2020.

[12] F. Hashimoto, Y. Guo, and A. Warren, “Surface integrity difference between hard turned and ground surfaces and its impact on fatigue life,” *CIRP Annals*, vol. 55, no. 1, pp. 81–84, 2006.

Vergleich modell- und datenbasierter prädiktiver Regelungsverfahren an einer nichtlinearen Regelstrecke

Marcel Zehner, Alessio Cavaterra, Steven Lambeck

FB Elektrotechnik und Informationstechnik, Hochschule Fulda
Leipziger Str. 123, 36037 Fulda
E-Mail: {marcel.zehner,alessio.cavaterra,steven.lambeck}@et.hs-fulda.de

1 Einführung

Die modellbasierte prädiktive Regelung (engl.: Model Predictive Control, MPC) hat in den letzten Jahren sowohl im akademischen als auch in der industriellen Praxis stetig an Bedeutung gewonnen. Der Ansatz basiert auf einem internen parametrischen Prozessmodell, welches aus gesammelten Daten des Prozesses identifiziert wird. Das im Regelbetrieb in jedem Abtastschritt gelöste Optimierungsproblem basiert auf diesem linearen Modell. In Kombination mit Takagi-Sugeno-Fuzzy-Systemen (TS-Systemen) kann trotz eines Regelgesetzes, welches ein lineares Modell nutzt, ein nichtlineares Gesamtübertragungsverhalten erreicht werden. Hierbei setzt sich das TS-System beispielsweise aus gewichteten lokal-linear-affinen Modellen zusammen, wodurch in Abhängigkeit des aktuellen Arbeitspunktes ein entsprechendes Prozessmodell generiert wird. Dadurch ist ein MPC mit TS-System für ein Vielzahl nichtlinearer Prozesse geeignet. Der Schritt der Systemidentifikation ist aber oft ein aufwändiger Prozess, der einen Großteil des Entwicklungsaufwandes eines modellbasierten Ansatzes beansprucht. Dies kann beispielsweise an einer falsch gewählten Modellstruktur liegen. Ebenso ist es möglich, dass das gewählte parametrische Modell eine hohe Modellierungsgüte aufweist, sich allerdings nicht für die Verwendung in Reglern eignet. Direkte datenbasierte prädiktive Regelungsverfahren, welche in den letzten Jahren enorm an Bedeutung gewonnen haben, umgehen den Schritt der Systemidentifikation zu Beginn des Reglerentwurfs. Sie nutzen Eingangs- und Ausgangsdaten des Prozesses als nicht parametrisches Modell, können damit Prädiktionen durchführen und im Zuge eines Optimierungsproblems die optimierte Stellgrößenfolge generieren. Ein Regelungsansatz, der genauso vorgeht und im vorliegenden Beitrag zum Vergleich herangezogen wird, ist Data Enabled Predictive Control [1]. An einem nichtlinearen strömungstechnischen Prozess, welcher in einem Labor des Fachbereichs Elektro- und Informationstechnik der Hochschule Fulda vorhanden ist, können beide Ansätze erprobt werden. Das Ziel des Beitrags ist es, den modell- und datenbasierten Ansatz mit Blick auf das Führungsverhalten zu vergleichen, Vor- und Nachteile sowie mögliche Erweiterungen um Methoden der Computational Intelligence für den datenbasierten Ansatz herauszuarbeiten. Gleichzeitig sollen dadurch zwei Regelungsansätze für den strömungstechnischen Prozess praktisch implementiert werden. Damit ist es möglich, den Studierenden in Veranstaltungen sowohl moderne prädiktive Regelungsansätze als auch Fuzzy Systeme näher zu bringen. Der Beitrag beginnt mit einer Beschreibung des Schwebekörperversuchsstandes. Danach werden die modell- und datenbasierten prädiktiven Regelungsmethoden erläutert. Im Anschluss werden die Ergebnisse der Regelungen gezeigt und im darauffolgenden Kapitel diskutiert.

2 Schwebekörperversuchsstand

Die Abbildung 1 zeigt den schematischen Aufbau des strömungstechnischen Schwebekörperversuchsstandes. Dieser wird für Lehr- und Forschungszwecke eingesetzt. Der Prozess besteht aus einer frei gelagerten Kreisscheibe, die innerhalb eines Plexiglaszylinders über einem Axiallüfter positioniert ist. Zur Vergrößerung der Geschwindigkeit des Luftstroms ist oberhalb des Axialüfters ein Konfusor (in Abbildung 1 rot dargestellt) verbaut. Die Kreisscheibe kann sich entlang der aus zwei Aluminiumstäben bestehenden Führung auf und ab bewegen. Im Regelkreis wird der durch den Axiallüfter erzeugte Luftstrom so eingestellt, dass sich die Scheibe auf die gewünschte Höhe bewegt und dort in einem schwebenden Zustand im Kräftgleichgewicht gehalten wird. Die Lüfterspannung u dient als Stellgröße und die Position y der Kreisscheibe stellt die Regelgröße dar. Die Regelgröße ist durch die maximale Höhe h des Plexiglaszylin-

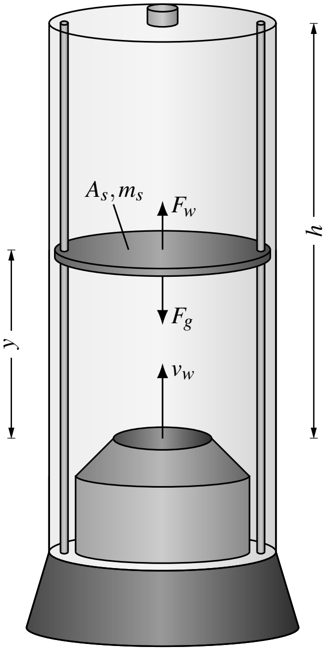


Bild 1: Schematische Darstellung des Schwebekörperversuchsstand

ders nach oben und das Ende des Konfusors nach unten beschränkt. Die Position y der Kreisscheibe wird mit Hilfe eines Ultraschallsensors erfasst, welcher am oberen Ende des Zylinders angebracht ist. Die theoretische Modellierung dieses Prozesses erfordert viel Zeit und einen hohen Simulationsaufwand. Das liegt an den nichtlinearen Strömungseffekten, wie beispielsweise Verwirbelungen und druckabhängigen Verlusten, die von der Position der Scheibe abhängen und nur mithilfe komplexer Simulationswerkzeuge der Computational Fluid Dynamics (CFD) näherungsweise gelöst werden können. Zudem treten an den Führungen der Scheibe nichtlineare Reibungseffekte auf, deren Modellierung ebenfalls mit erheblichem Aufwand verbunden ist. Die im System wirkenden Kräfte sind die durch den Luftstrom des Lüfters wirkende Kraft F_w sowie die Gewichtskraft F_g. Dadurch ergibt sich die folgende Differentialgleichung [2]:

$$m_s \cdot \ddot{y} = \frac{1}{2} \cdot C_d \cdot \rho \cdot A_s \cdot (v_w - \dot{y})^2 - m_s \cdot g \tag{1}$$

Dabei ist m_s die Masse der Kreisscheibe und A_s die Fläche der Kreisscheibe, welche dem Luftstrom des Lüfters ausgesetzt ist und g die Fallbeschleunigung. C_d beschreibt den dimensionslosen Luftwiderstandsbeiwert des im Luftstrom befindlichen Objektes und ρ ist in diesem Fall die Luftdichte, in dem sich die Kreisscheibe bewegt. v_w ist die Geschwindigkeit des Luftstroms innerhalb des Zylinders und y ist die Position der Scheibe. $\dot{y}$ bzw. $\ddot{y}$ ist dementsprechend die Geschwindigkeit bzw. Beschleunigung der Scheibe. Hier ist auch die Nichtlinearität der Differentialgleichung zu erkennen, weil die Differenz zwischen v_w und $\dot{y}$ quadratisch in die Gleichung mit einfließt. An diesem Versuchsstand werden modell- sowie datenbasierte prädiktive Regelstrategien erprobt, welche in den kommenden zwei Abschnitten genauer untersucht werden.

3 Modellbasierte prädiktive Regelung

3.1 Parametrische Systemdarstellung

Üblicherweise werden in modellprädiktiven Regelungsansätzen parametrische Modellstrukturen verwendet. In diesem Beitrag wird ein NARX-Modellansatz herangezogen, welcher mit Hilfe eines TS-Systems [3,4] und dem LOLIMOT-Konstruktionsalgorithmus [5] umgesetzt wird.

$$\underline{\hat{y}}_{\mathrm{TS}}(k) = \sum_{i=1}^{n_m} \Phi_i(\underline{z}) \underline{\theta}_{\mathrm{TS},i} \underline{x} \, . \tag{2}$$

Die Regressoren sind im Vektor $\underline{x}$ zusammengefasst und beinhalten die Eingangsgrößen und Ausgangsgrößen der jeweils zwei letzten Abtastschritte: die Lüfterspannung $u(k-1)$ und $u(k-2)$ sowie die Position der Kreisscheibe $y(k-1)$ und $y(k-2)$. Hierdurch wird die Identifikation von zeitdiskreten Übertragungsfunktionen zweiter Ordnung ermöglicht. Der Vektor $\underline{\theta}_{\mathrm{TS},i}$ beinhaltet die Parameter des i-ten ARX-Teilmodells. Die Fuzzy-Basisfunktionen $\Phi_i(\underline{z})$ werden aus dem gewichteten Mittel der Zugehörigkeitsgrade berechnet.

$$\Phi_i(\underline{z}) = \frac{\mu_i(\underline{z})}{\sum\limits_{i=1}^{n_m} \mu_i(\underline{z})} \quad , \quad \sum_{i=1}^{n_m} \Phi_i(\underline{z}) = 1 \, . \tag{3}$$

Deren Wertebereich ist demnach auf das Intervall $I = [0,1]\ I \in \mathbb{R}$ beschränkt. Die $\mu_i(\underline{z})$ Zugehörigkeitsfunktionen (ZF) stellen Gaußglocken dar. Eine einzelne Gauß'sche ZF ist über ihr Zentrum ν und die Standardabweichung σ vollständig bestimmt:

$$\mu_{\text{Gauss}}(x) = \exp\left(\frac{-(x-\nu)^2}{2\sigma^2}\right) . \tag{4}$$

Die Schedulingvariable $\underline{z}$ ist ein Vektor, der die Lüfterspannung $u(k-1)$ und die Höhe der schwebenden Kreisscheibe $y(k-1)$ des jeweils letzten Abtastschritts enthält. Der bereits erwähnte LOLIMOT-Algorithmus wird zur achsenorthogonalen Einteilung der Prämisse des TS-Modells genutzt. Die gewählte Aufteilung ist in Abbildung 2 zu sehen. Das TS-Modell besteht aus insgesamt sechs Teilmodellen.

3.2 Nichtlineare modellprädiktive Regelung mit mehreren Teilmodellen

Ein wesentlicher Vorteil der Formulierung in Gleichung (2) ist, dass die linear-affinen Teilmodelle zu jedem Abtastschritt über das Fuzzy-Regelwerk miteinander gewichtet kombiniert werden können und somit in jedem Abtastschritt ein linear-affines Modell berechnet werden kann. Das nichtlineare Übertragungsverhalten kann somit über ein im aktuellen Abtastschritt gültiges lineares-affines Modell approximiert werden. Letzteres wird anschließend im Optimalsteue-

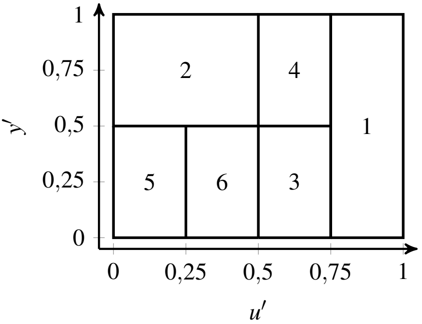


Bild 2: Prämisse des TS-Modells (normiert). Die Teilmodelle sind nummeriert.

rungsproblem (5) genutzt, um eine optimale Stellgrößentrajektorie für den zu regelnden Prozess zu berechnen.

$$\begin{array}{ll} \underset{u,y}{\text{minimize}} & \sum_{k=0}^{N_P-1}\left(\|y_k-r\|_Q^2+\|u_k\|_R^2+\|\Delta u_k\|_{R_\Delta}^2\right) \\ \text{subject to} & u_k\in\mathcal{U},\forall k\in\{0,\dots,N_U-1\}, \\ & y_k\in\mathcal{Y},\forall k\in\{0,\dots,N_P-1\}. \end{array} \tag{5}$$

$\mathcal{U}$ bezeichnet hierbei die Menge zulässiger Stellwerte, welche in diesem Fall alle Werte innerhalb $0\,\mathrm{V}\leq u_k\leq 10\,\mathrm{V}$ beschreibt. Die Kreisscheibe kann nur innerhalb der Höhen $0\,\mathrm{cm}\leq y_k\leq h$ bewegt werden, sodass diese Werte Teil der Menge $\mathcal{Y}$ sind. Die Abtastzeit beträgt 100 ms, der Prädiktionshorizont $N_P=70$ und der Kontrollhorizont $N_U=4$. Die Wichtungsmatrizen werden gewählt zu $Q=0.6$, $R=0$ und $R_\Delta=0.5$. Der hier umgesetzte Regler wird im weiteren Verlauf als TSMPC bezeichnet.

4 Datenbasierte prädiktive Regelung

4.1 Nicht parametrische Systemdarstellung

Im Gegensatz zu den modellbasierten prädiktiven Regelungsansätzen nutzen direkte datenbasierte prädiktive Regelungsmethoden keine parametrischen Modellstrukturen, wie beispielsweise lineare Zustandsraummodelle oder ARX-Modelle, sondern einen nicht parametrischen Ansatz für das interne Modell der Regelstrecke. Diese nicht parametrische Systemdarstellung baut auf der *behavioral* Systemtheorie [6] und dem *Fundamental Lemma nach Willems* [57] auf. Diese Grundlage erlaubt, für das folgende lineare zeitinvariante System (6)

$$\begin{aligned} x_{k+1} &= Ax_k+Bu_k, \\ y_k &= Cx_k+Du_k, \end{aligned} \tag{6}$$

mit dem Zustand $x_k\in\mathbb{R}^n$, dem Eingang $u_k\in\mathbb{R}^m$ und dem Ausgang $y_k\in\mathbb{R}^p$ eine nicht parametrische Systemdarstellung basierend auf Eingangs- und Ausgangsdaten von (6). Die gesammelten Eingangs- und Ausgangsdaten werden hierbei in Form von Hankelmatrizen angeordnet, wobei die Hankelmatrizen das nicht parametrische Modell darstellen. Für eine Eingangsdatentrajektorie $u = \{u_i\}_{i=1}^{T_d}$ der Länge T_d, wobei $T_d \geq L$, ergibt sich eine Hankelmatrix mit L Reihen und $T_d - L + 1$ Spalten:

$$\mathscr{H}_L(u) := \begin{pmatrix} u_1 & u_2 & \cdots & u_{T_d-L+1} \\ u_2 & u_3 & \cdots & u_{T_d-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ u_L & u_{L+1} & \cdots & u_{T_d} \end{pmatrix}. \tag{7}$$

Das Eingangssignal u wird *persistently exciting* der Ordnung L genannt, wenn die Hankelmatrix $\mathscr{H}_L(u)$ vollen Reihenrang besitzt. Der Begriff *persistently exciting* beschreibt ein Eingangssignal, welches das System ausreichend anregt und dadurch Ausgangsdaten erzeugt, welche das typische Systemverhalten repräsentieren [1]. Für Eingangs- sowie Ausgangsdaten der Form $(u_d, y_d) = \{u_{d_i}, y_{d_i}\}_{i=1}^{T_d}$ von (6), wobei der Index d für in der Vergangenheit aufgenommene Daten steht, mit der Länge T_d und einem Eingangssignal $u_d = \{u_{d_i}\}_{i=1}^{T_d}$, dessen Hankelmatrix $\mathscr{H}_L(u_d)$ vollen Reihenrang besitzt, gehören die Eingangs- und Ausgangstrajektorien $(u, y) = \{u_i, y_i\}_{i=1}^{L}$ zu (6) dann und nur dann, wenn ein $g \in \mathbb{R}^{T_d-L+1}$ existiert, sodass:

$$\begin{pmatrix} \mathscr{H}_L(u_d) \\ \mathscr{H}_L(y_d) \end{pmatrix} g = \begin{pmatrix} u \\ y \end{pmatrix}. \tag{8}$$

Dieses Ergebnis erlaubt es, für ein lineares zeitinvariantes System jede mögliche Trajektorie aus einer endlichen Anzahl vergangener Trajektorien zu konstruieren. Hierbei werden die vergangenen Trajektorien, welche in der Hankelmatrix angeordnet sind, linear miteinander kombiniert, wodurch sie eine mögliche Trajektorie des Systems ergeben. g kann dabei, als Wichtungsvektor für die vergangenen Trajektorien in der Hankelmatrix gesehen werden. Mit dieser Formulierung ist es möglich, mit Hilfe von Eingangs- sowie Ausgangsdaten und ohne parametrisches Modell, das Systemverhalten zu prädizieren. Dafür werden die Hankelmatrizen in (8) aufgeteilt in:

$$\begin{pmatrix} U_\mathrm{p} \\ U_\mathrm{f} \end{pmatrix} := \mathscr{H}_{T_\mathrm{ini}+N_P}(u_d), \quad \begin{pmatrix} Y_\mathrm{p} \\ Y_\mathrm{f} \end{pmatrix} := \mathscr{H}_{T_\mathrm{ini}+N_P}(y_d). \tag{9}$$

$U_p \in \mathbb{R}^{T_{ini} \times T_d - N_P - T_{ini} + 1}$ besteht hierbei aus den ersten T_{ini} Reihen der Hankelmatrix $\mathscr{H}_{T_{\mathrm{ini}}+N_P}(u_d)$ und $U_f \in \mathbb{R}^{N_P \times T_d - N_P - T_{ini} + 1}$ setzt sich aus den nächsten N_P Reihen der Hankelmatrix $\mathscr{H}_{T_{\mathrm{ini}}+N_P}(u_d)$ zusammen. Die Zusammensetzung von Y_P und Y_F erfolgt analog. Im weiteren Verlauf werden $(u_{ini}, y_{ini}) = \{u_{k-i}, y_{k-1}\}_{i=T_{ini}}^{1}$ als die neusten Eingangs- sowie Ausgangsdaten des Systems definiert. Mit dieser Definition sowie den Hankelmatrizen U_p, Y_p, U_f, Y_f lässt sich (8) erweitern zu:

$$\begin{pmatrix} U_{\mathrm{p}} \\ Y_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{f}} \end{pmatrix} g = \begin{pmatrix} u_{\mathrm{ini}} \\ y_{\mathrm{ini}} \\ u \\ y \end{pmatrix} \tag{10}$$

Die ersten beiden Gleichungen in (10) werden dafür genutzt, um den impliziten Zustand des Systems zu schätzen, von wo aus die Prädiktion stattfinden kann [8]. Mit den neusten Eingangs- und Ausgangsdaten (u_{ini}, y_{ini}), den Hankelmatrizen der linken Seite von Gleichung (10) sowie einem gegebenen Eingangssignal u kann unter Berücksichtigung der ersten drei Gleichungen von (10) g bestimmt werden. Die datengetriebene Prädiktion des Ausgangssignal findet mit Hilfe der Gleichung $y = Y_f g$ statt. Gleichung (10) kann somit für eine implizite Zustandsschätzung und gleichzeitige Prädiktion verwendet werden, womit es als nicht parametrisches Modell in prädiktiven Regelungsansätzen zum Einsatz kommen kann.

4.2 Data Enabled Predictive Control

Bei Data Enabled Predictive Control (DeePC) handelt es sich um einen prädiktiven Regelungsansatz, welcher lediglich auf Eingangs- und Ausgangsdaten des zu regelnden Systems basiert [1]. Wie bereits erwähnt, ersetzt Gleichung (10) das parametrische Modell, wie es in modellbasierten Ansätzen zu finden ist. Unter dieser Kenntnis ergibt sich das folgende Optimierungsproblem, welches in jedem Abtastschritt der Regelung neu gelöst wird:

$$
\begin{array}{ll}
\underset{g,u,y}{\text{minimize}} & \sum_{k=0}^{N_P-1}\left(\|y_k-r\|_Q^2+\|u_k\|_R^2\right)\\
\text{subject to} & \begin{pmatrix} U_\text{p}\\ Y_\text{p}\\ U_\text{f}\\ Y_\text{f}\end{pmatrix} g = \begin{pmatrix} u_\text{ini}\\ y_\text{ini}\\ u\\ y\end{pmatrix},\\
& u_k \in \mathcal{U}, \forall k \in \{0,\dots,N_P-1\},\\
& y_k \in \mathcal{Y}, \forall k \in \{0,\dots,N_P-1\}.
\end{array}
\tag{11}
$$

Das Optimierungsproblem in (11) eignet sich für lineare zeitinvariante Systeme ohne Messrauschen. N_P ist hierbei der Prädiktionshorizont, T_{ini} beschreibt die Länge der Initialisierungstrajektorien u_{ini} und y_{ini}. Die Kostenfunktion betrachtet die Abweichung der Regelgröße y von der Führungsgröße r, sowie die Stellgröße u_k. Die Abweichung von der Führungsgröße wird mit der Wichtungsmatrix $Q \in \mathbb{R}^{p\times p}$ bewertet, u hingegen wird mit der Wichtungsmatrix $R \in \mathbb{R}^{m\times m}$ bemessen. Bei N_P, T_{ini} sowie Q und R handelt es sich um Hyperparameter des Reglers. Gleichzeitig sind Beschränkungen für die Regel- und Stellgröße, in Form der Mengen $\mathcal{Y}$ und $\mathcal{U}$ im Optimierungsproblem vorhanden. Da die datenbasierte prädiktive Regelung auf dem Schwebekörperversuchsstand umgesetzt werden soll, wo Messrauschen und Nichtlinearitäten vorkommen, muss das Optimierungsproblem in (11) angepasst werden. Hierbei schlägt [1] die Verwendung von Regularisierungstermen und Slack-Variablen vor, um Messrauschen sowie Nichtlinearitäten zu begegnen. Dabei entsteht das folgende Optimierungsproblem:

$$
\begin{array}{ll}
\underset{g,\Delta u,y,\sigma_y,\sigma_{\Delta u}}{\text{minimize}} & \sum_{k=0}^{N_P-1}\left(\|y_k-r\|_Q^2+\|\Delta u_k\|_R^2\right)\\
& +\|g\|_{\lambda_g}^2+\left\|\sigma_y\right\|_{\lambda_y}^2+\|\sigma_{\Delta u}\|_{\lambda_{\Delta u}}^2\\
\text{subject to} & \begin{pmatrix} \Delta U_\text{p}\\ Y_\text{p}\\ \Delta U_\text{f}\\ Y_\text{f}\end{pmatrix} g = \begin{pmatrix} \Delta u_\text{ini}\\ y_\text{ini}\\ \Delta u\\ y\end{pmatrix} + \begin{pmatrix} \sigma_{\Delta u}\\ \sigma_y\\ 0\\ 0\end{pmatrix},\\
& u_k \in \mathcal{U}, \forall k \in \{0,\dots,N_P-1\},\\
& \Delta u_k \in \Delta\mathcal{U}, \forall k \in \{0,\dots,N_P-1\},\\
& y_k \in \mathcal{Y}, \forall k \in \{0,\dots,N_P-1\}.
\end{array}
\tag{12}
$$

Zusätzlich zu der Kostenfunktion, welche die Abweichung zur Führungsgröße sowie den Stelleingriff bestraft, wird die Optimierungsvariable g in der Kostenfunktion minimiert und mit λ_g gewichtet. Da bedingt durch Messrauschen die Gleichheitsnebenbedingung in (11) nicht erfüllt werden können, werden die Slackvariablen σ_y sowie $\sigma_{\Delta u}$ eingeführt, um die Lösbarkeit des Optimierungsproblems in jedem Abtastschritt zu gewährleisten. Beide Slackvariablen werden in der Kostenfunktion mit einem entsprechenden Gewicht λ_y sowie $\lambda_{\Delta u}$ bewertet. Dadurch ergeben sich bei der Reglerimplementierung zusätzliche Tuningparameter. Gleichzeitig wird das Ergebnis aus [9] in (12) umgesetzt. Dabei wird anstatt der Stellgröße u, die Stellgrößenrate Δu verwendet. Um die Stellgrößenrate zu verwenden, werden zusätzlich die Hankelmatrizen U_p sowie U_f durch ΔU_p und ΔU_f ersetzt, welche die Änderung der Stellgröße betrachten. Ähnlich wie bei modellbasierten Ansätzen [10] wird dadurch ein Integral-Anteil zum Regelgesetz ergänzt, wodurch wir eine bleibende Regelabweichung eliminieren können. Die Implementierung eines DeePC läuft so ab, dass zu Beginn Daten (u_d, y_d) gesammelt werden, wobei das Eingangssignal u_d *persistently exciting* ist. Auf Basis der Eingangs- und Ausgangsdaten werden die Hankelmatrizen gebildet, welche im Optimierungsproblem (12) Verwendung finden. Nach geeigneter Wahl der Reglerparameter wird (12) in jedem Abtastschritt gelöst. Vor der eigentlichen Regelung des Schwebekörperversuchsstandes wird der Prozess mit einem Amplitude Modulated Pseudo Random Binary Signal (APRBS) für 90 s angeregt. Danach werden mit Hilfe der entstandenen Eingangs- und Ausgangsdaten die Hankelmatrizen in (12) besetzt. Die Beschränkungen der Stellgröße liegen bei $0\,\mathrm{V} \leq u_k \leq 10\,\mathrm{V}$, die der Regelgröße bei $0\,\mathrm{cm} \leq y_k \leq h$. Für den Regler werden folgende Werte verwendet, wobei die Tuningparameter des Reglers mit Hilfe einer Rastersuche gefunden wurden:

$$\begin{aligned} &T_d = 450,\ N_P = 30,\ T_{ini} = 8,\ \lambda_g = 5 \cdot 10^{-4}, \\ &\lambda_y = 1000,\ \lambda_u = 1 \cdot 10^{-4},\ Q = 8,\ R = 0.01 \end{aligned} \tag{13}$$

Der Regler arbeitet mit einer Abtastzeit von 200 ms, da das Optimierungsproblem mit $T_d - N_P - T_{ini} + 1 = 413$ Optimierungsvariablen im Vergleich zum TSMPC größer ist und somit in den 100 ms Abtastzeit des TSMPC nicht lösbar ist. Für die Lösung des quadratischen Programms findet die Verwendung des DAQP-Solvers [11] statt.

5 Vergleich der Ergebnisse

Am Versuchsaufbau wird das Führungsverhalten des TSMPC und DeePC anhand mehrerer Sollwertsprünge untersucht. Die Sollwerttrajektorie hat eine Länge von 500 s und variiert im Bereich von 10 cm bis 35 cm. Die Sollwerttrajektorie beginnt mit einem Sprung von 0 cm auf 30 cm. Im Anschluss verringert sich der Sollwert alle 100 s um 10 cm bis eine Führungsgröße von 10 cm erreicht ist. Daraufhin erfolgen zwei Sollwertsprünge um jeweils 15 cm, womit am Ende ein Sollwert von 35 cm vorliegt.

Die Abbildung 3 zeigt den Sollwertverlauf, den jeweiligen Regelgrößenverlauf sowie die generierte Stellgröße der beiden Regelungsansätze. Beide Methoden sind in der Lage, den Prozess auf die gewünschten Sollwerte zu regeln. Hierbei weist der TSMPC beim ersten Sollwertsprung eine Anregelzeit von 5 s auf. Nach einem Überschwingen regelt der TSMPC nach etwa 32 s auf den Sollwert von 30 cm ein. Im Gegensatz dazu, hat der DeePC eine Anregelzeit von 42 s und nähert sich von unten allmählich dem Sollwert an. Generell ist zu beobachten, dass der DeePC bei allen Sollwertsprüngen eine größere Anregelzeit aufweist. Bei dem Sollwertsprung von 30 cm auf 20 cm zeigt der rein datenbasierte Ansatz ein längeres Überschwingen, kann im Anschluss aber auf den Sollwert regeln. Beim Sollwertsprung von 20 cm auf 10 cm schwingt der vom DeePC geregelte Prozess um den Sollwert. Im Anschluss ist die Regelung auf 10 cm aber möglich. Der TSMPC hingegen regelt mit einer geringeren An- bzw. auch Ausregelzeit auf den vorgegebenen Sollwert, wobei bei den Sollwertwechseln 20 cm auf 10 cm und von 10 cm auf 25 cm ein leichtes Überschwingen zu beobachten ist. Die beiden darauffolgenden Sollwerte können von den Reglern mit einer ähnlichen Regelgenauigkeit erreicht werden, wobei die Anregelzeit des TSMPC erneut kleiner ist. Die Betrachtung der Stellgröße u bestätigt das Verhalten, indem zu sehen ist, dass der TSMPC aggressiver in den Prozess eingreift. Dagegen zeigt der DeePC kein starkes Überschwingen der Stellgröße zu Beginn eines jeden Sollwertwechsels, was auf eine konservative Reglerauslegeung deutet.

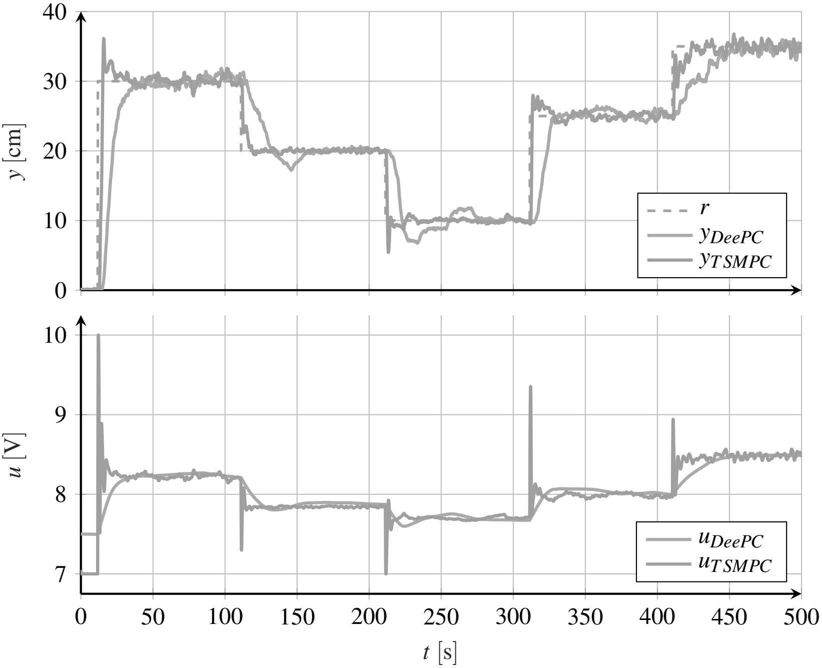


Bild 3: Ergebnisse des TSMPC und DeePC

6 Diskussion und Ausblick

Wie im vorherigen Abschnitt 5 gezeigt wurde, eignen sich beide prädiktiven Ansätze dafür, die nichtlineare strömungstechnische Regelstrecke zu regeln. Der modellbasierte Ansatz nutzt dabei ein TS-System als Prozessmodell, welches sich aus gewichteten lokal-linear-affinen Modellen zusammensetzt. Dadurch kann der TSMPC über einen weiten Arbeitsbereich eine zufriedenstellende Regelgüte erzielen. Der datenbasierte Ansatz umgeht eine Modellbildung und nutzt lediglich Eingangs- und Ausgangsdaten, welche ein nicht parametrisches Modell darstellen. Beide Ansätze können ebenso im Rahmen von Lehrveranstaltungen genutzt werden, um den Studierenden die Grundlagen sowohl moderner prädiktiver Regelungsansätze (modellbasiert und datenbasiert) als auch von Fuzzy-Systemen zu vermitteln. Der DeePC zeigt, in den unteren Arbeitsbereichen ein Überschwingen, was in einer schlechteren Regelgüte resultiert. Ein Grund hierfür sind die mangelnden Daten in diesem Bereich. Das gewählte Anregungssignal für den datenbasierten Ansatz hat hauptsächlich Bereiche oberhalb von 15 cm angeregt. Dadurch fehlen Eingangs- und Ausgangsdatentrajektorien, welche die Dynamik des Prozesses in diesem Bereich beschreiben. Außerdem ist der hier gezeigte Regler in (12) ein linearer Regler, was zusätzlich die Performance-Verluste bei der Regelung einer nichtlinearen Regelstrecke erklärt. In der Literatur gibt es jedoch einige Vorschläge zum Umgang mit nichtlinearen Regelstrecken bei der Verwendung von datenbasierten Ansätzen. Dabei werden die Hankelmatrizen mit den letzten Eingangs- und Ausgangsdaten erneuert. In jedem Abtastschritt wird die erste Reihe der Hankelmatrix in (7) gelöscht, alle anderen Reihen werden um eins nach oben versetzt und es entsteht eine neue letzte Reihe der Hankelmatrix, welche die aktuellen Daten beinhaltet. Der interessierte Leser wird auf [9], [13] und [14] verwiesen. Eine weitere Möglichkeit ist die Nutzung von TS-Systemen. Dabei könnte das TS-System in Abhängigkeit des aktuellen Arbeitspunktes entscheiden, welche Hankelmatrizen dem datenbasierten Regelungsansatz übergeben werden. Dies wird in zukünftigen Arbeiten genauer untersucht. Gleichzeitig ist der DeePC weniger dynamischer als der TSMPC und weist eine geringere Anregelzeit vor. Dies ist darin begründet, dass die Gewichte entsprechend konservativ gewählt werden mussten, weil ansonsten der durch den DeePC geregelte Prozess ein zu starkes Überschwingen bei Sollwertwechseln zeigt. Generell hat sich die Wahl der Reglerparameter als langwierig herausgestellt. Für zukünftige Reglereinstellungen kann die Anwendung evolutionärer Algorithmen, wie z. B. des Differential-Evolutions-Algorithmus, genutzt werden, um die Reglerparameter zu optimieren. Gleichzeitig ist zu erwähnen, dass in den Hankelmatrizen lediglich 90 s des nichtlinearen Prozess betrachtet werden, wodurch nur ein gewisser Teil des dynamischen Verhaltens erfasst werden kann. Mit einer Vergrößerung des Anregungszeitraums könnte dies verhindert werden, allerdings geht dies mit einer Vergrößerung von T_d einher, was zu einem Optimierungsproblem mit mehr Optimierungsvariablen und einer längeren Berechnungszeit führen würde. Dadurch wäre das gestellte Optimierungsproblem in der gegebenen Abtastzeit von 200 ms nicht mehr lösbar. Eine Möglichkeit, dieser Problematik zu begegnen, ist der Ansatz in [15], welcher in zukünftigen Arbeiten umgesetzt wird.

Literatur

[1] J. Coulsen, J. Lygeros und F. Dörfler. „Data Enabled Predictive Control: In the Shallows of the DeePC“. .18th European Control Conference (ECC), S. 307-312. 2019.

[2] J. Chacon et al. „Design of a Low-Cost Air Levitation System for Teaching Control Engineering“. Sensors. 2017.

[3] O. Nelles, „Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models“. Springer. 2020.

[4] A. Kroll,„Computational Intelligence. Probleme, Methoden und technische Anwendungen.“ De Gruyter Studium. 2016.

[5] B. Hartmann, T. Ebert, T. Fischer, T.Belz, J. Kampmann und O. Nelles. „LMNtool – Toolbox zum automatischen Trainieren lokaler Modellnetze“. *22. Workshop Computational Intelligence, KIT Scientific Publishing*. 45, S. 341–355. 2012.

[6] J. W. Polderman und J.C. Willems „Introduction to mathematical systems theory: a behavioral approach“. Springer-Verlag. 1997.

[7] J.C.Willems, P. Rapisarda, I. Markovsky und B.L.M. De Moor. „A note on persistence of excitation“. Systems & Control Letters, 54(4), S. 312-329. 2005.

[8] I. Markovsky und P. Rapisarda „Data-driven simulation and control“. International Journal of Control, 81(12), S. 1946-1959. 2008.

[9] M. Lazar und P.C.N. Verheijen „Offset-free data-driven predictive control“ IEEE 61st Conference on Decision and Control (CDC), S.1099-1104. 2022

[10] J. Maciejowski „Predictive Control with Constraints“. Prentice Hall, England. 2022.

[11] D. Arnström, A. Bemporad und D. Axehill „A Dual Active-Set Solver for Embedded Quadratic Programming Using Recursive LDL^T Updates“. IEEE Transactions on Automatic Control, 67, S.4362 - 4369. 2022.

[12] J. Berberich et al. „Data-driven model predictive control with stability and robustness guarantees“. IEEE Transactions on Automatic Control 66(4), S.4406-4421 2022.

[13] S. Baros et al. „Online data-enabled predictive control“. Automatica, 138 2022.

[14] J. Teutsch et al. „An Online Adaption Strategy for Direct Data-driven Control“. IFAC-PapersOnLine, Vol. 56, S.644-649. 2023.

[15] K. Zhang et al. „Dimension Reduction for Efficient Data-Enabled Predictive Control“. IEEE Control Systems Letters, Vol. 7, S.3277 - 3282. 2023.

An Application-oriented Review of Standard and Integral Sparse Identification of Nonlinear Dynamics

Mortimer Dockhorn, Martin Kohlhase

Institute for Data Science Solutions, Hochschule Bielefeld
Interaktion 1, 33619 Bielefeld
E-Mail: {mortimer.dockhorn, martin.kohlhase}@hsbi.de

1 Introduction

The aim of this article is to provide an application-oriented overview of two variants of the Sparse Identification of Nonlinear Dynamics (SINDy). The original variant introduced by Brunton et al. in [1], and a modified integral version presented by Schaeffer and McCalla in [2]. The SINDy algorithm is, in general, a method for estimating the governing equations of nonlinear dynamical systems from measurement data. It is based on sparse regression and therefore designed to find a parsimonious mathematical model with the fewest terms necessary to accurately describe the underlying dynamics of a system. SINDy leverages techniques from the field of data science to produce interpretable models that can provide additional insight into the fundamental principles that describe a system's behavior. Data-driven methods for system identification are also valuable in scenarios where more traditional approaches, such as modeling from first principles, are impractical due to the high complexity of a system or because the governing scientific laws are not fully known. These techniques can therefore accelerate industrial tasks, such as controller design, and support research by providing dynamic models that offer new insights into a process.

As presented in [1], SINDy is capable of estimating a nonlinear discrete-time model in the form of difference equations as well as a nonlinear continuous-time model in the form of ordinary differential equations (ODEs). This work, however, focuses solely on continuous-time models, as ODEs are likely to provide a better basis for an in-depth analysis of a system. This is partly due to the fact that the form and parameter values of difference equations are influenced by factors such as the discretization method or the sample time, whereas the attributes of ODEs are determined solely by the system they describe. ODEs, therefore, provide more accessible information when governing physical laws or the physical interpretation of parameters are of interest. For more information on the estimation of discrete-time models with SINDy, the reader is referred to [1, 3].

The nonlinear ODE model generated by SINDy is of the general form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \tag{1}$$

where $\boldsymbol{u}(t)$, $\boldsymbol{x}(t)$, $\dot{\boldsymbol{x}}(t)$, and $\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$ are vectors whose elements represent potential inputs to the system, the system's state variables, the state variables' first derivatives with respect to time, and the corresponding state equations. This model is estimated from training data for $\boldsymbol{u}(t)$, $\boldsymbol{x}(t)$, and $\dot{\boldsymbol{x}}(t)$ by approximating the right-hand side of (1) as linear combinations of user-defined candidate functions. The construction of training datasets therefore requires data of all the inputs, state variables, and state variables' first derivatives with respect to time. Especially the last requirement, however, creates challenges since the time derivatives of a system's state variables are not always measurable, and common numerical differentiation techniques (e.g., finite differences) are likely to produce corrupted derivative approximations when applied to noisy measurements of the state variables. To address this issue, more advanced methods, such as the Total Variation Regularized Derivative [4], Spline Polynomial Interpolation [5], and an algorithm based on the Savitzky-Golay Filter [6] were proposed in [1, 7–9], [10], and [11], respectively. While these methods produce significantly more accurate derivative approximations of noise-contaminated data, some of them, like the Savitzky-Golay Filter, can corrupt the data, and therefore the derivative, by altering curvature or moving extrema. Other methods, such as the Total Variation Regularized Derivative, may be less prone to altering the data but have limitations on the size of the dataset they can be applied to. To address these issues, an algorithm for batchwise Tikhonov Regularized Differentiation (TRD) is presented in this work. The algorithm extends the basic TRD [12], which has high accuracy compared to many of the aforementioned methods [5], but is also limited with regard to the size of the dataset it can be applied to. Batchwise TRD splits larger datasets into batches for the differentiation and merges the individual segments back into a single derivative, allowing it to be applied to datasets of all sizes. This combination of high accuracy and flexibility makes batchwise TRD a suitable tool for generating training data for SINDy when only noisy measurements of a system's state variables are available. For a more detailed description of most of the differentiation methods referenced in this work, the reader is referred to [5].

An alternative variant of SINDy, that does not require the state variables' derivatives for the construction of training datasets, was introduced in [2]. This variant will be referred to as integral SINDy (I-SINDy) within this work, as the key difference between SINDy and I-SINDy is given by the fact that I-SINDy uses the time integral of relationship (1) for the model estimation. This modified approach also translates to the training data, as the time integrals of the state variables are required for the construction of training datasets instead of their time derivatives. While these integrals are also not measurable in many cases, computing them from noisy measurement data of the state variables poses notably less challenges then it does for the derivatives. This is due to the fact that even simple numerical integration schemes resemble a high robustness against noise. Moreover, numerical integration is performed iteratively and therefore not constraint with regard to the dataset size. I-SINDy and SINDy with batchwise TRD can therefore be recognized as alternative options for the identification of nonlinear ODE models, when only noisy measurements of a system's states are available. This leads to the aim of this article, which is providing an application-oriented comparison of both methods.

The remainder of the article is structured as follows. The general SINDy framework and the modifications resulting in I-SINDy are presented together with the batchwise TRD algorithm in section 2. The robustness of both methods to noise in the training data as well as the results that were achieved with data of a real world tank system, are illustrated in section 3. The results are summarized in section 4 and future research directions are discussed.

2 The SINDy, I-SINDy, and TRD Methods

This section provides a detailed overview of the SINDy and I-SINDy methods, as well as the batchwise TRD algorithm. The general framework of SINDy is illustrated in 2.1, and the modifications leading to I-SINDy are summarized in 2.2. Subsequently, the batchwise TRD algorithm is be presented in 2.3.

2.1 The SINDy Method

Since SINDy was originally introduced in [1], the general concepts discussed here are drawn from that source unless stated otherwise.

SINDy estimates a nonlinear ODE model of the the general form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \tag{1}$$

where $\boldsymbol{u}(t) \in \mathbb{R}^p$ and $\boldsymbol{x}(t)$, $\dot{\boldsymbol{x}}(t)$, $\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \in \mathbb{R}^n$ are column vectors whose elements represent the inputs to the system, the system's state variables, the state variables' first derivatives with respect to time, and the corresponding state equations. The model is estimated from training data for $\boldsymbol{u}(t)$, $\boldsymbol{x}(t)$ and $\dot{\boldsymbol{x}}(t)$ by approximating the individual state equations in terms of of k user-defined candidate functions $g_i : Y_i \rightarrow Z_i,\ i \in \{1, 2, \ldots, k\}$. Assuming the training data is real-valued, $g_i(\cdot)$ is a scalar and real-valued function of w_i scalar and real-valued arguments. Formally, this equates to $Z_i \subseteq \mathbb{R}$, and $Y_i := Y_{i,1} \times Y_{i,2} \cdots \times Y_{i,w_i}$ with $Y_{i,j} \subseteq \mathbb{R}$ for all $j \in \{1, 2, \ldots, w_i\}$. The model estimation process yields n coefficient vectors $\boldsymbol{v}_i := [v_{i,1} \ \ldots \ v_{i,k}]^{\mathrm{T}} \in \mathbb{R}^k$, $i \in \{1, 2, \ldots, n\}$, with each vector determining the contribution of the candidate functions to a corresponding state equation. The estimated model is therefore composed of n state equations, with each being a linear combination of the candidate functions.

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} v_{1,1}g_1(\cdot) + v_{1,2}g_2(\cdot) \cdots + v_{1,k}g_k(\cdot) \\ v_{2,1}g_1(\cdot) + v_{2,2}g_2(\cdot) \cdots + v_{2,k}g_k(\cdot) \\ \vdots \qquad\qquad \vdots \qquad \ddots \qquad \vdots \\ v_{n,1}g_1(\cdot) + v_{n,2}g_2(\cdot) \cdots + v_{n,k}g_k(\cdot) \end{bmatrix} \tag{2}$$

For the construction of training datasets, the data for $\boldsymbol{u}(t)$, $\boldsymbol{x}(t)$, and $\dot{\boldsymbol{x}}(t)$ are required. Since this section focuses on the general framework of SINDy, it will be assumed that these data are available in sufficiently high quality and that they are given as discrete measurements collected at the time instances $t_1, t_2, \dots, t_m$.

A training dataset for SINDy consists of three data matrices and the vector $\boldsymbol{t} := [t_1 \dots t_m]^{\mathrm{T}} \in \mathbb{R}^m$. To construct the matrices, the measurements are arranged into the column vectors μ_i, χ_j, $\dot{\chi}_j \in \mathbb{R}^m$, with $i \in \{1, 2, \dots, p\}$ and $j \in \{1, 2, \dots, n\}$, where each vector holds the data for one input, state variable, or time derivative, respectively.

$$\mu_i := \begin{bmatrix} u_i(t_1) \\ \vdots \\ u_i(t_m) \end{bmatrix}, \quad \chi_j := \begin{bmatrix} x_j(t_1) \\ \vdots \\ x_j(t_m) \end{bmatrix}, \quad \dot{\chi}_j := \begin{bmatrix} \dot{x}_j(t_1) \\ \vdots \\ \dot{x}_j(t_m) \end{bmatrix}$$

Subsequently, the matrices $\boldsymbol{U} \in \mathbb{R}^{m \times p}$ and $\boldsymbol{X}$, $\dot{\boldsymbol{X}} \in \mathbb{R}^{m \times n}$ are constructed from these vectors and provided to SINDy as training data along with $\boldsymbol{t}$.

$$\boldsymbol{U} := \begin{bmatrix} \mu_1 & \dots & \mu_p \end{bmatrix}, \quad \boldsymbol{X} := \begin{bmatrix} \chi_1 & \dots & \chi_n \end{bmatrix}, \quad \dot{\boldsymbol{X}} := \begin{bmatrix} \dot{\chi}_1 & \dots & \dot{\chi}_n \end{bmatrix}$$

In addition to the training data, SINDy requires the user to define the candidate functions. This step is analogous to the tuning of an essential hyperparameter, as the quality of the estimated model largely depends on it. In general, any function with a domain and codomain compatible with the training data can be used as a candidate function. In practice, the selection of candidate functions should be based on the characteristics of the system being modeled. For example, if the system exhibits oscillatory behavior, trigonometric functions such as sine and cosine may be a reasonable choice. Another common choice is powers of the inputs or state variables, as they often appear in models that were derived from first principles. A constant bias, as well as functions that depend solely on time, may also be selected if the corresponding behavior is observed in the system.

Provided with the training data and candidate functions, SINDy constructs the library matrix

$$\boldsymbol{D} := \begin{bmatrix} g_1(\cdot)\big|_{t=t_1} & g_2(\cdot)\big|_{t=t_1} & \cdots & g_k(\cdot)\big|_{t=t_1} \\ g_1(\cdot)\big|_{t=t_2} & g_2(\cdot)\big|_{t=t_2} & \cdots & g_k(\cdot)\big|_{t=t_2} \\ \vdots & \vdots & \ddots & \vdots \\ g_1(\cdot)\big|_{t=t_m} & g_2(\cdot)\big|_{t=t_m} & \cdots & g_k(\cdot)\big|_{t=t_m} \end{bmatrix} \in \mathbb{R}^{m\times k}.$$

The structure of **D** is such that each column contains the values of a candidate function evaluated at the time instances in **t**.

Following the construction of the library matrix, SINDy estimates the individual state equations. Within this process, Sparse Regression Methods (SRMs) are employed for solving the optimization problems that yield the coefficient vectors. SRMs are used in in particular, as they are designed to balance model accuracy and complexity. Within the context of SINDy, this translates to the objective of finding a set of state equations, where each is composed of the fewest number of candidate functions necessary to accurately describe the dynamics of the training data. The use of SRMs, therefore, counteracts overfitting and prevents unnecessary model complexity. Moreover, the results in [1] suggest that sparse models are more likely to represent the actual physical laws that govern a system's behavior. Examples for SRMs, that have been used particularly within the SINDy framework, are the Sequential Threshold Least Squares algorithm [1], the Douglas-Rachford algorithm [2, 13], the Least Absolute Shrinkage and Selection Operator [11, 14], and Sparse Relaxed Regularized Regression [15, 16]. As the referenced literature suggests, that all of these methods are compatible with the SINDy framework, the choice of a particular SRM is not as important as, for instance, the choice of candidate functions. However, some methods may still perform better in specific scenarios than others (e.g., limited training data or high noise levels). It is therefore suggested, to perform an initial comparison of multiple models, where each is estimated with a different SRM. This approach is feasible without to much additional effort, as an open source software package for SINDy ([3]) is available, that allows the user to choose from different pre-implemented SRMs. Since the specifics of the aforementioned sparse regression algorithms are beyond the scope of this article, a general approach for the estimation of the state equations will be illustrated in the remainder of this section.

SINDy performs the model estimation in n steps, corresponding to the estimation of n state equations. In each step, the selected SRM aims to recover a sparse coefficient vector that provides the best fit to the training data. Independent of the particular choice of SRM, the overall objective of the estimation process may be expressed as

$$\underset{\boldsymbol{v}_i \in S}{\operatorname{argmin}}(\dot{\chi}_i - \boldsymbol{D}\boldsymbol{v}_i), \quad i \in \{1, 2, \ldots, n\}. \tag{3}$$

where S is the set of all sparse coefficient vectors. It should be noted however, that (3) is only used for conveying the general aim of the optimization, as the particular form of the optimization problem depends on the choice of SRM.

Subsequent to the calculation of all coefficient vectors, SINDy constructs the row vector

$$\delta^{\mathrm{T}}(t) := \begin{bmatrix} g_1(\cdot) & g_2(\cdot) & \ldots & g_k(\cdot) \end{bmatrix} \in \mathbb{R}^k,$$

whose elements are the symbolic candidate functions, arranged in the same order as the corresponding data in $\boldsymbol{D}$. It should be noted, that $\boldsymbol{D}$ contains discrete data values that were generated by applying the candidate functions to the training data, while $\delta^{\mathrm{T}}(t)$ contains the actual symbolic candidate functions.

In the final step, SINDy recovers the state equations via the relationship

$$\dot{x}_i(t) = \delta^{\mathrm{T}}(t)\boldsymbol{v}_i \quad i \in \{1, 2, \ldots, n\}.$$

Which is equivalent to the form (2), that was used in the introduction of this section.

2.2 The I-SINDy method

This subsection illustrates the modifications to the general SINDy framework that result in I-SINDy. It therefore builds strongly on the content of 2.1, where the general SINDy framework is presented. Accordingly, 2.1 is recommended as preliminary literature. As I-SINDy was originally introduced in [1], the general concepts illustrated here are drawn from that source unless stated otherwise.

I-SINDy also estimates a nonlinear ODE model of the general form (1). For the model estimation process, however, I-SINDy uses the relationship

$$\boldsymbol{x}(t)-\boldsymbol{x}(t_1)=\int_{t_1}^{t}\boldsymbol{f}(\boldsymbol{x}(\tau),\,\boldsymbol{u}(\tau),\,\tau)\,\mathrm{d}\tau, \tag{4}$$

which is the time integral of (1), assuming measurements begin at $t = t_1$. Equivalent to SINDy, I-SINDy approximates the right-hand side of (4) as linear combinations of candidate functions. However, the derivatives of the candidate functions are first defined in terms of $\boldsymbol{x}(t)$, $\boldsymbol{u}(t)$, and t and then numerically integrated. Thus, the construction of training datasets requires the measurement data to be integrated rather than differentiated.

Provided with the training data and choice of candidate functions, I-SINDy constructs the library matrix $\boldsymbol{D}$. Since this process is identical for SINDy and I-SINDy, the reader is referred to section 2.1, where a detailed description is provided. As I-SINDy estimates the integrals of the state equations, the numerical integrals of the columns of $\boldsymbol{D}$ are used for the estimation. This ensures that the state equations of the final ODE model are given as linear combinations of the selected candidate functions and thus makes it easier to chose candidate functions based on knowledge of the system being modeled. With $\boldsymbol{D} = [\boldsymbol{d}_1,\, \boldsymbol{d}_2, \ldots,\, \boldsymbol{d}_k]$, I-SINDy constructs the the integral library matrix

$$\boldsymbol{D}^{\text{int}} := \begin{bmatrix}\boldsymbol{d}_1^{\text{int}} & \boldsymbol{d}_2^{\text{int}} & \ldots & \boldsymbol{d}_k^{\text{int}}\end{bmatrix} \in \mathbb{R}^{m\times k},$$

where $\boldsymbol{d}_i^{\text{int}} \in \mathbb{R}^m$ is the column vector of data values that was acquired by numerical cumulative integration of the column $\boldsymbol{d}_i$. Subsequently, I-SINDy performs the estimation of model (4) with the Data in $\boldsymbol{D}^{\text{int}}$. As the remaining estimation process is mostly identical to the process within SINDy, the remainder of this subsection will only provide a brief summary.

Similar to SINDy, I-SINDy aims to recover sparse coefficient vectors. The general objective of the optimization is just a slightly modified version of (3).

$$\underset{\boldsymbol{v}_i \in S}{\operatorname{argmin}}(\widetilde{\chi}_i - \boldsymbol{D}^{\text{int}}\boldsymbol{v}_i), \quad i \in \{1,2,\ldots,n\}.$$

Instead of derivative data, the vector $\widetilde{\chi}_i := \chi_i - x_i(t_1)\mathbf{1}_m \in \mathbb{R}^m$, is used in the optimization, which modifies the data in χ_i to represent the integral of $\dot{\chi}_i$, with the condition $\boldsymbol{x}(t_1) = 0$. This step is required since the numerical integration causes the first value of all $\boldsymbol{d}_i^{\text{int}}$, $i \in \{1,\ 2,\ \ldots,\ k\}$ to be zero.

Equivalent to SINDy, the desired ODE model (1) can be composed from the state equations

$$\dot{x}_i(t) = \boldsymbol{\delta}^{\mathrm{T}}(t)\,\boldsymbol{v}_i,\ \ i \in \{1,\ 2,\ \ldots,\ n\}.$$

Here $\boldsymbol{\delta}^{\mathrm{T}}(t)$ is a row vector that contains the symbolic candidate functions corresponding to the columns of $\boldsymbol{D}$ (not $\boldsymbol{D}^{\text{int}}$).

2.3 Batchwise Tikhonov Regularized Differentiation

This subsection introduces the batchwise TRD algorithm. First, the concept of standard TRD is illustrated through an example implementation proposed in [5], and subsequently the batchwise TRD algorithm is presented.

Given a column vector $\boldsymbol{y} := [y_1\ y_2\ \ldots\ y_n]^{\mathrm{T}} \in \mathbb{R}^n$ that contains discrete data, collected at the corresponding time instances $t_1, t_2, \ldots, t_n$, TRD defines the L_2-regularized optimization problem

$$\underset{\dot{\boldsymbol{y}}}{\operatorname{argmin}} ||\boldsymbol{A}\dot{\boldsymbol{y}} - \widetilde{\boldsymbol{y}}||_2^2 + k||\boldsymbol{B}\dot{\boldsymbol{y}}||_2^2. \tag{5}$$

Solving (5) yields the column vector $\dot{\boldsymbol{y}} \in \mathbb{R}^n$, whose elements approximate the first time derivative of the data in $\boldsymbol{y}$ at the time instances $t_1, t_2, \ldots, t_n$. To provide a foundation for understanding this concept, the individual elements in (5) are defined in the following passage.

The matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is constructed such that calculating the product $\boldsymbol{A}\dot{\boldsymbol{y}}$ is equivalent to applying the cumulative trapezoidal rule to $\dot{\boldsymbol{y}}$. Assuming a uniform sample time $\Delta t := t_{i+1} - t_i$ for all $i \in \{1,2,\dots,n-1\}$, this yields

$$\boldsymbol{A} := \frac{1}{2}\Delta t \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 2 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 2 & \dots & 1 \end{bmatrix}. \tag{6}$$

Therefore, if $\boldsymbol{y}^{\text{trap}} := \boldsymbol{A}\dot{\boldsymbol{y}} = [y_1^{\text{trap}} \ \dots \ y_n^{\text{trap}}]^{\text{T}} \in \mathbb{R}^n$, then y_i^{trap} approximates the definite integral from t_1 to t_i of a function that is continuous on $[t_1, t_i]$ and interpolates the first i data values in $\boldsymbol{y}$. Additionally, since (6) is defined such that $y_1^{\text{trap}} = 0$ for any given $\boldsymbol{y}$, the column vector $\widetilde{\boldsymbol{y}} := \boldsymbol{y} - y_1 \mathbf{1}_n \in \mathbb{R}^n$ is used in (5), as $\widetilde{y}_1 = 0$ is guaranteed by definition.

The matrix $\boldsymbol{B} \in \mathbb{R}^{(n-2)\times n}$ is defined such that calculating the product $\boldsymbol{B}\dot{\boldsymbol{y}}$ is equivalent to applying the central difference method to $\dot{\boldsymbol{y}}$ while discarding the edge values. The scalar $k \in \mathbb{R}_{>0}$ is a hyperparameter that determines the amount of regularization.

$$\boldsymbol{B} := \frac{1}{2\Delta t} \begin{bmatrix} -1 & 0 & 1 & 0 & \dots & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 0 & 1 \end{bmatrix}$$

Given these definitions, the term $||\boldsymbol{A}\dot{\boldsymbol{y}} - \widetilde{\boldsymbol{y}}||_2^2$ in (5) can be interpreted as the requirement to find a $\dot{\boldsymbol{y}}$ whose approximated integral matches the data $\widetilde{\boldsymbol{y}}$ as closely as possible. The term $||\boldsymbol{B}\dot{\boldsymbol{y}}||_2^2$, on the other hand, can be interpreted as the requirement that the values of the approximated first derivative of $\dot{\boldsymbol{y}}$ are as small as possible. This latter requirement reduces high-frequency noise in $\dot{\boldsymbol{y}}$, as $||\boldsymbol{B}\dot{\boldsymbol{y}}||_2^2$ would become large when such noise is present. The combination of both requirements results in the objective of finding a derivative approximation that is accurate while containing minimal noise. The tuning of k can be interpreted as adjusting the relative weight of each requirement in the overall objective. In particular, large k prioritize smoothness, whereas small k emphasize accuracy.

Even though a general analytical solution to (5) is given by

$$\dot{\boldsymbol{y}} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + k\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}\boldsymbol{A}^{\mathrm{T}}\widetilde{\boldsymbol{y}}, \tag{7}$$

numerical solvers are preferred for solving (5) because they are more robust than the numerical matrix inversion required in (7). In practice, however, constraints on memory and computational resources make TRD unsuitable for large $\boldsymbol{y}$, as (5) is posed as a single optimization problem. To address this issue, the following batchwise TRD algorithm is proposed, which splits larger datasets into batches for the differentiation and merges the individual segments back into a single derivative.

3 Results

This section illustrates the results of the performed comparison between SINDy and I-SINDy. In 3.1, the sensitivity of both variants to measurement noise is compared using noise-contaminated simulation datasets from two different models. In 3.2, both methods are applied to the data of a real tank system, and the quality of the estimated models is compared.

3.1 Robustness to Noise

In this subsection, two models are considered. The Lotka-Volterra model, which describes the dynamics of interacting predator and prey populations, and the Duffing model, which describes a nonlinear oscillator. The governing equations of both models are listed in table 1. Simulation data for each model is generated, using scipy's solve_ivp method with a Runge-Kutta 45 solver. A time span of 10 seconds is simulated, with a uniform step size of 1 ms. This results in each simulation dataset containing 10^3 values per state. For convenience in notation

Algorithm 1.: Batchwise TRD

```
Voraussetzung: Dataset to be differentiated: $data$, number of samples that the batches
    should overlap: $overlap$, uniform sample time: $\Delta t$, regularization parameter: $k$,
    tolerance value for merging derivative batches: $tol$
Abschlussbedingung: An array containing the derivative values: $reg\_derivative$
function TRD($\boldsymbol{y}$, $k$, $\Delta t$)
    $n \leftarrow$ Number of elements in $\boldsymbol{y}$
    $\boldsymbol{A}, \boldsymbol{B} \leftarrow$ Initialize matrices of size $(n, n)$ and $(n-2, n)$
    $\widetilde{\boldsymbol{y}} \leftarrow$ Subtract first element of $\boldsymbol{y}$ from all elements of $\boldsymbol{y}$
    for $i = 0$ to $n-2$ do
        $\boldsymbol{A}[i+1, : i+2] \leftarrow \left[\frac{1}{2}, 1, \frac{1}{2}\right] \cdot \Delta t$
    Set $\boldsymbol{B}$: main diagonal $(-2\Delta t)^{-1}$, and 2nd diagonal $(2\Delta t)^{-1}$
    $\dot{\boldsymbol{y}} \leftarrow$ Solve least square: $[\boldsymbol{A}, \sqrt{k}\boldsymbol{B}]^{\mathrm{T}} \dot{\boldsymbol{y}} = [\widetilde{\boldsymbol{y}}, \boldsymbol{0}]^{\mathrm{T}}$
    return $\dot{\boldsymbol{y}}$
Initialize empty derivative_list, set data_len and batch_num to 0
while batch_num $<$ $\lceil$data_len/batchsize$\rceil$ do
    start_idx $\leftarrow$ batch_num $\cdot$ batchsize $-$ overlap $\cdot$ (batch_num $> 0$)
    end_idx $\leftarrow$ min(start_idx $+$ batchsize $+$ overlap, data_len)
    Append TRD(data[$start\_idx : end\_idx$], $k$, $\Delta t$) to derivative_list
    batch_num++
for $i = 0$ to length(derivative_list) $- 2$ do
    Find overlap between derivative_list[i] and derivative_list[i+1]
    Trim derivative_list[i] and append relevant part of derivative_list[i+1]
Set reg_derivative to the final merged batch
```

$q := 10^3$ within this section. The simulation data is then contaminated with zero-mean normally distributed noise and subsequently used as training data for both SINDy variants. This yields the general relationship $\boldsymbol{x}_i^{\text{train}} := \boldsymbol{x}_i^{\text{sim}} + \boldsymbol{d}_i$, where $\boldsymbol{x}_i^{\text{train}}$, $\boldsymbol{x}_i^{\text{sim}} \in \mathbb{R}^q$ are vectors containing the training and simulation data for the i-th state variable, respectively. The vector $\boldsymbol{d}_i := [d_i(t_1)\ d_i(t_2)\ \dots d_i(t_q)] \in \mathbb{R}^q$ contains the corresponding noise values, which are drawn from a normal distribution.

$$d_i(t) \sim \mathcal{N}(0, \sigma_i^2)$$

Table 1: Models used for generating the training and test data

Model	Governing Equations
Lotka-Volterra	$\dot{x}_1(t) = 1.5x_1(t) - x_1(t)x_2(t)$ $\dot{x}_2(t) = x_1(t)x_2(t) - 3x_2(t)$
Duffing	$\dot{x}_1(t) = x_2(t)$ $\dot{x}_2(t) = -0.2x_2(t) - x_1(t) - x_1^3(t)$

The intensity of the noise relative to the simulation data is defined in terms of the noise level NL, which itself is defined as the ratio of the energies of the noise and the simulation data.

$$\mathrm{NL} := \frac{\|\boldsymbol{d}_i\|_2^2}{\|\boldsymbol{x}_i^{\mathrm{sim}}\|_2^2} = \frac{\sigma_i^2}{\|\boldsymbol{x}_i^{\mathrm{sim}}\|_2^2}$$

Since all noise values are drawn from a zero-mean normal distribution, the energy of the noise, that is added to the i-th state variable, can be approximated by the variance σ_i^2. This relationship is used for the simulation data of each state variable individually, to calculate the variance that yields a desired noise level. In particular, 4 noise levels are considered for the comparison. The intensities of these levels relative to the simulation data are illustrated in figures provided in the Appendix: Supplementary Figures for Section 3.

$$\mathrm{NL} \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$$

For the comparison of both SINDy variants, 4 training datasets, corresponding to the 4 noise levels are constructed for each model in table 1. These datasets are provided to the SINDy software package [3], and the standard and weak form integral variants are used. All models are estimated with the Sequential Threshold Least Squares optimizer, and the derivatives required by SINDy are approximated with batchwise TRD. The training data are not preprocessed or smoothed in any way. In all scenarios, integer powers of the state variables up to the tenth power and products of both state variables, with one variable also

Table 2: Initial conditions used for the generation of training and test datasets.

Model	Training data	Test data
Lotka-Volterra	$x_1(t_0) = 10$ $x_2(t_0) = 5$	$x_1(t_0) = 1$ $x_2(t_0) = 10$
Duffing	$x_1(t_0) = 5$ $x_2(t_0) = 2$	$x_1(t_0) = 1$ $x_2(t_0) = 4$

going up to the tenth power, are used as candidate functions. A bias is also included in the library. The accuracy of each model is then measured through cross validation. Test datasets are generated by creating simulation data with different initial conditions than used for the training data. All initial conditions employed in this comparison are listed in table 2. The estimated models are simulated with identical solver settings as those used for the training and test data. The accuracy of the models is quantified using the root mean squared error ε between the model trajectories and the noise free training and test data, respectively. This yields

$$\varepsilon_i^{\text{train}} := \frac{\|\boldsymbol{x}_i^{\text{sim}} - \hat{\boldsymbol{x}}_i\|_2}{\sqrt{q}}, \qquad \varepsilon_i^{\text{test}} := \frac{\|\boldsymbol{x}_i^{\text{test}} - \hat{\boldsymbol{x}}_i\|_2}{\sqrt{q}}.$$

Here $\boldsymbol{x}_i^{\text{test}}$, $\hat{\boldsymbol{x}}_i \in \mathbb{R}^q$ are vectors, containing the test and the model data for the i-th state variable, respectively. For each model, only the highest error value in the state variables is considered for the evaluation of the model accuracy. Since both models in table 1 have two state variables, the following errors are defined

$$\varepsilon\text{-Train} := \max(\varepsilon_1^{\text{train}},\ \varepsilon_2^{\text{train}}),$$
$$\varepsilon\text{-Test} := \max(\varepsilon_1^{\text{test}},\ \varepsilon_2^{\text{test}}).$$

Another metric used to quantify the accuracy of the estimated models is the Maximum Coefficient Error MCE. For a general definition of the MCE, the absolute coefficient error Δc^{abs} is first defined for a model with o coefficients.

$$\Delta c_i^{\text{abs}} := \left|c_i^{\text{ref}} - c_i^{\text{mod}}\right|, \quad i \in \{1, 2, \ldots, o\}$$

Here c_i^{ref} is the i-th coefficient of the reference model that was used for the generation of training and test data, and c_i^{mod} is the corresponding coefficient of the estimated model. Given this definition, the MCE is then defined as

$$\text{MCE} := \frac{\max(\Delta c_1^{\text{abs}},\, \Delta c_2^{\text{abs}}, \ldots, \Delta c_o^{\text{abs}})}{\left|c_{\text{max}}^{\text{mod}}\right|} \cdot 100\,\%. \tag{8}$$

The term $c_{\text{max}}^{\text{mod}}$ refers to the coefficient of the reference model, which has the biggest absolute coefficient error. If, for example, the numerator of (8) yields Δc_2^{abs}, then $c_{\text{max}}^{\text{mod}} = c_2^{\text{mod}}$.

The introduced metrics for all estimated models are listed in the tables 3 through 6, where each table holds the results for one noise level. The additional column "Structure" holds information about whether the structure of the reference model could be recovered. A "No" entry is therefore equivalent to a failed system identification. For these cases no metrics are listed, as some of the incorrect models where unstable or completely unable to match the dynamics of the reference model. In Addition, supplementary figures, which illustrate the trajectories of the training and test data as well as the achieved model accuracy are provided in the Appendix: Supplementary Figures for Section 3.

Table 3: Results with NL $= 10^{-4}$

Model	Method	Structure	MCE	ε-Train	ε-Test
L.-V.	SINDy	Yes	0.5 %	0.38	0.57
	I-SINDy	Yes	0.2 %	< 0.01	0.03
Duffing	SINDy	Yes	10.3 %	0.26	0.11
	I-SINDy	Yes	0.2 %	0.19	0.02

Table 4: Results with NL = 10^{-3}

Model	Method	Structure	MCE	ε-Train	ε-Test
L.-V.	SINDy	Yes	2.7 %	1.70	2.49
	I-SINDy	Yes	0.3 %	0.03	0.04
Duffing	SINDy	Yes	23.7 %	0.33	0.31
	I-SINDy	Yes	1.8 %	0.05	0.02

Table 5: Results with NL = 10^{-2}

Model	Method	Structure	MCE	ε-Train	ε-Test
L.-V.	SINDy	No	–	–	–
	I-SINDy	Yes	0.2 %	0.03	0.04
Duffing	SINDy	Yes	158 %	2.86	1.69
	I-SINDy	Yes	33 %	0.87	0.28

Table 6: Results with NL = 10^{-1}

Model	Method	Structure	MCE	ε-Train	ε-Test
L.-V.	SINDy	No	–	–	–
	I-SINDy	Yes	3.0 %	1.75	2.32
Duffing	SINDy	No	–	–	–
	I-SINDy	No	–	–	–

3.2 Tank System

In this subsection, SINDy and I-SINDy are used on measurement data collected from a real-world tank system, as it is shown in figure 1. Two scenarios are considered. In the first, the drain valve connected to tank two is open, and the flow valve connecting the tanks two and three is closed. In this case the system

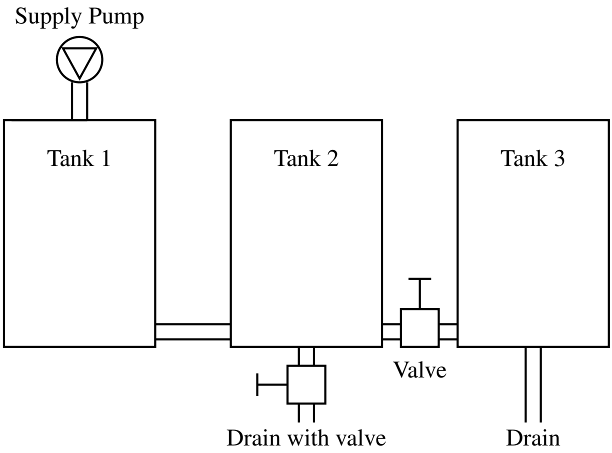


Figure 1: Schematic of the tank system that measurement data was taken from. If the drain valve from tank two is opened and the flow valve connecting tanks 2 and 3 is closed, then the third tank is inactive, yielding a two-tank system. If the drain valve from tank two is closed and the flow valve is opened, then the system operates as three tank system.

operates as a two tank system. In the second scenario, the drain valve from tank two is closed and the flow valve between tanks two and three is open. In this case, all three tanks are active. In both scenarios, the supply pump is always active, providing a continuous inflow into the the first tank. The state variables of the system are the water levels in the tanks. From measurements of the water levels, two training datasets are constructed. One for the operation as two tank system, and another for the operation as three tank system. Each dataset contains measurements sampled with a uniform step size of 1 s. Both datasets contain approximately 1,400 values for each state variable. The training datasets are provided to the SINDy software package, and the standard as well as weak form integral variant are used for estimating a model. Since a model based on Torricelli's law is commonly used for describing similar systems, terms from this model are added to the library. In particular,

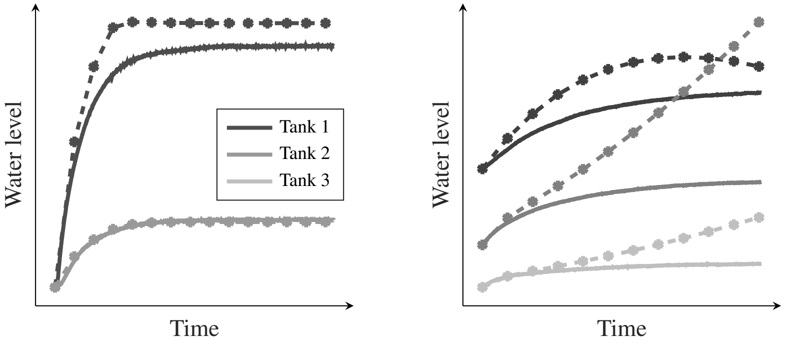
$$\begin{aligned}
&\sqrt{h_i(t)}, \\
&\operatorname{sign}(h_i(t) - h_j(t)) \cdot \sqrt{\left|h_i(t) - h_j(t)\right|}, \\
&\text{with } i, j \in \{1, 2, 3\},\ i \neq j,
\end{aligned}$$

where $h_1(t), h_2(t)$, and $h_3(t)$ are water levels in the corresponding tanks. In addition, integer powers of the state variables up to the tenth power and products of both state variables, with one variable also going up to the tenth power, are used as candidate functions. A bias, as well as the sine and cosine of the state variables are also included in the library.

The training data and the achieved model quality are illustrated in figure 2. Only I-SINDy was able to estimate a model that resembles the general dynamics of the two tank system (see fig 2a). For the three tank system, I SINDy was not able to estimate a stable model, as it could not be simulated for the time span of the training data. SINDy was not able to estimate an applicable model for either configuration. The water levels simulated with SINDy's two tank model did not deviate from the initial conditions, and the three tank model shows seemingly unstable behavior (see fig. 2b). The two tank model estimated by I-SINDy is given by

$$\begin{aligned}\dot{h}_1(t) =& -0.164\sqrt{h_1(t)} + 0.132\sqrt{h_2(t)} + 6.035\sin(h_1(t))\ldots\\ & -160.315\sin(h_2(t)) + 0.016\cos(h_1(t)) - 5.750h_1(t)\ldots\\ & +159.828h_2(t),\\ \dot{h}_2(t) =& 0.002\sqrt{h_1(t)} - 9.322\sin(h_2(t)) + 9.300h_2(t).\end{aligned}$$

Even tho this model is able to qualitatively match the dynamics in the training data, the model structure is notably more complicated than for models of equivalent tank systems, that can be found in the literature. Moreover, the presence of the sine and cosine terms in both state equations, does not seem to depict any underlying physical laws, as no oscillatory behavior is observed in the tank system. The complicated model structure as well as the presence of several trigonometric terms seems to indicate overfitting.



(a) Two-tank system (tank 3 inactive)

(b) Three-tank system

Figure 2: Training data (solid) and simulated trajectories (star-dash) for the two-tank system (a) and the three-tank system (b). The simulated trajectories in (a) were generated from the model estimated by SINDy, and in (b) from the model estimated by I-SINDy. The legend in (a) applies to both figures.

4 Summary and Discussion

The results acquired with the synthetic datasets have shown, that weak form integral SINDy is significantly more robust against noise in the training data than standard SINDy. Even tho batchwise TRD was used, weak form I-SINDy showed a higher estimation accuracy at high noise levels. These results were observed for the Duffing and the Lotka-Volterra model, with weak form I-SINDy being able to estimate relatively accurate models at noise levels where SINDy was not able to correctly identify the structure of the model.

While I-SINDy also yielded better results for tank system data, none of both methods was able to estimate an applicable model. This result shows, that it may not always be possible to create the conditions, which are needed for the general SINDy framework to perform reliably. Especially the choice of candidate functions poses a significant limitation, since the structure and relevant terms of a model, that is to be identified, are not always known. Contributions, that provide an alternative to manually selecting the library terms, do therefore have a great potential for making the general SINDy framework more suitable for real world applications.

References

[1] Steven L. Brunton, Joshua L. Proctor and J. Nathan Kutz. “Discovering governing equations from data by sparse identification of nonlinear dynamical systems”. In: *Proceedings of the National Academy of Sciences* 113.15 pp. 3932-3937. 2016. https://doi.org/10.1073/pnas.1517384113

[2] Hayden Schaeffer and Scott G. McCalla. “Sparse model selection via integral terms”. In: *Phys. Rev. E* 96.2:023302. 2017. https://doi.org/10.1103/PhysRevE.96.023302

[3] Alan A. Kaptanoglu et al.. “PySINDy: A comprehensive Python package for robust sparse system identification”. In: *Journal of Open Source Software* 7.69. 2022. https://doi.org/10.21105/joss.03994

[4] Rick Chartrand. “Numerical Differentiation of Noisy, Nonsmooth Data”. In: *International Scholarly Research Notices* 2011 pp. 1-11. 2011. https://api.semanticscholar.org/CorpusID:12989196

[5] Ian Knowles and Robert J. Renka. “Methods for numerical differentiation of noisy data”. In: *Electronic Journal of Differential Equations* Conference 21, pp. 235-246. 2014. https://ejde.math.txstate.edu/conf-proc/21/k3/knowles.pdf

[6] Abraham Savitzky and M. J. E. Golay. “Smoothing and Differentiation of Data by Simplified Least Squares Procedures”. In: *Analytical Chemistry* 36.8 pp. 1627-1639. 1964. https://doi.org/10.1021/ac60214a047

[7] N. M. Mangan, S. L. Brunton, J. L. Proctor and J. N. Kutz. “Inferring Biological Networks by Sparse Identification of Nonlinear Dynamics”. In: *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications* 2.1 pp. 52-63. 2016. https://doi.org/10.1109/TMBMC.2016.2633265

[8] E. Kaiser, J. N. Kutz and S. L. Brunton. “Sparse identification of nonlinear dynamics for model predictive control in the low-data limit”. In: *Proc. R. Soc. A* 474:20180335. 2018. https://doi.org/10.1098/rspa.2018.0335

[9] Kadierdan Kaheman, J. Nathan Kutz and Steven L. Brunton. “SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics”. In: *Proc. R. Soc. A* 476:20200279. 2020. https://doi.org/10.1098/rspa.2020.0279

[10] Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor and J. Nathan Kutz. “Data-driven discovery of partial differential equations”. In: *Science Advances* 3.4. 2017. https://doi.org/10.1126/sciadv.1602614

[11] K. Egan, W. Li and R. Carvalho. “Automatically discovering ordinary differential equations from data with sparse regression”. In: *Commun. Phys.* 7:20. 2024. https://doi.org/10.1038/s42005-023-01516-2

[12] Jane Cullum. “Numerical Differentiation and Regularization”. In: *SIAM Journal on Numerical Analysis* 8.2 pp. 254-265. 1971. https://doi.org/10.1137/0708026

[13] Patrick L. Combettes and Jean-Christophe Pesquet. “Proximal Splitting Methods in Signal Processing”. In: *Fixed-Point Algorithms for Inverse Problems in Science and Engineering* (Springer New York) pp. 185-212. 2011. https://doi.org/10.1007/978-1-4419-9569-8_10

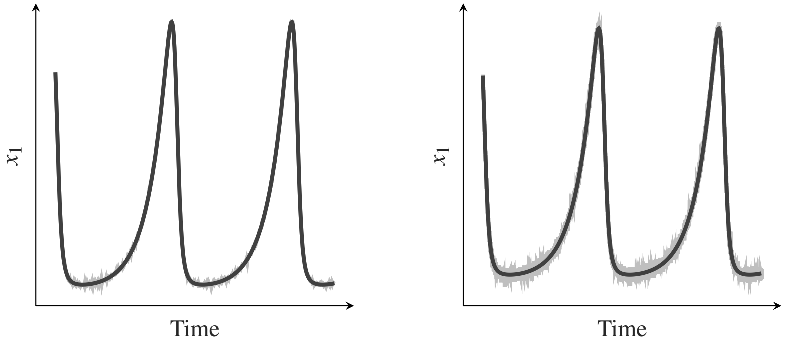
[14] Robert Tibshirani. “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 pp. 267-288. 1996. https://www.jstor.org/stable/2346178

[15] Yifei Guan, Steven L. Brunton and Igor Novosselov. “Sparse nonlinear models of chaotic electroconvection”. In: *R. Soc. Open Sci* 8:202367. 2021. https://doi.org/10.1098/rsos.202367

[16] Peng Zheng et al.. “A Unified Framework for Sparse Relaxed Regularized Regression: SR3”. In: *IEEE Access* 7 pp. 1404-1423. 2019. https://doi.org/10.1109/ACCESS.2018.2886528

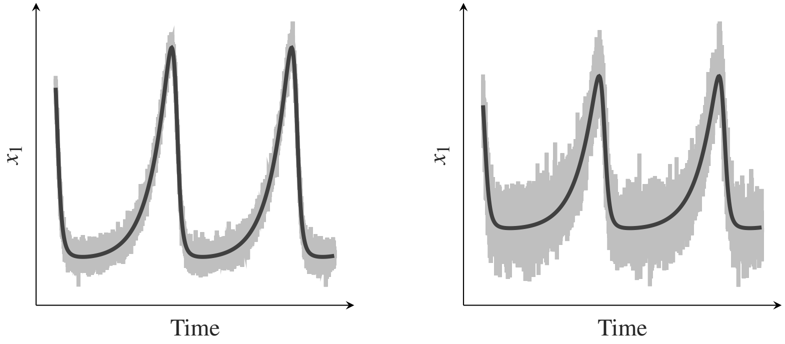
Appendix: Supplementary Figures for Section 3

Appendix



(a) Noise level = 10^{-4}

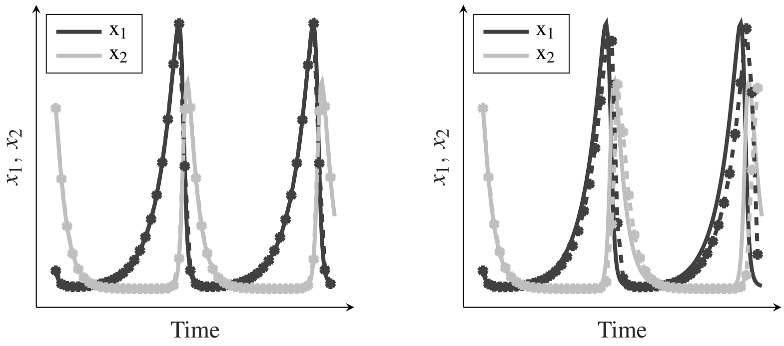
(b) Noise level = 10^{-3}



(c) Noise level = 10^{-2}

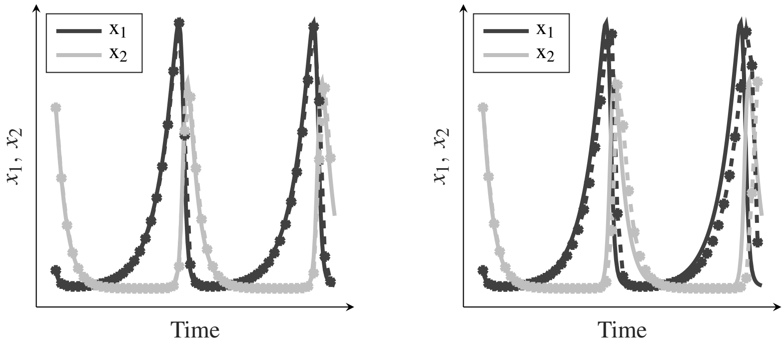
(d) Noise level = 10^{-1}

Figure 3: Data corresponding to state variable x_1 from the Lotka-Volterra model. The initially generated simulation data is shown in black and the added noise in light gray. The sum of both signals is used as training data.



(a) I-SINDy, noise level $= 10^{-2}$

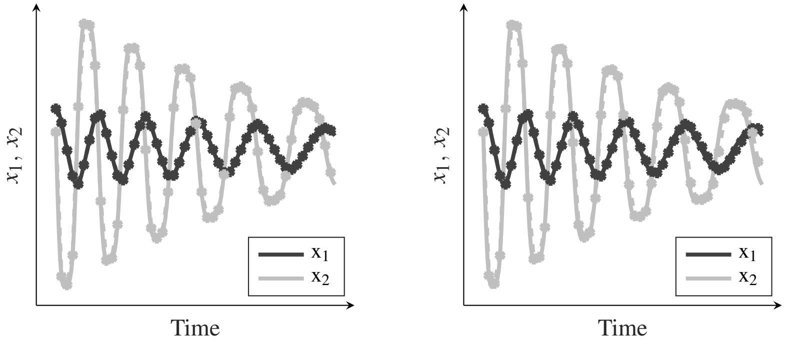
(b) I-SINDy, noise level $= 10^{-1}$



(c) SINDy, noise level $= 10^{-4}$

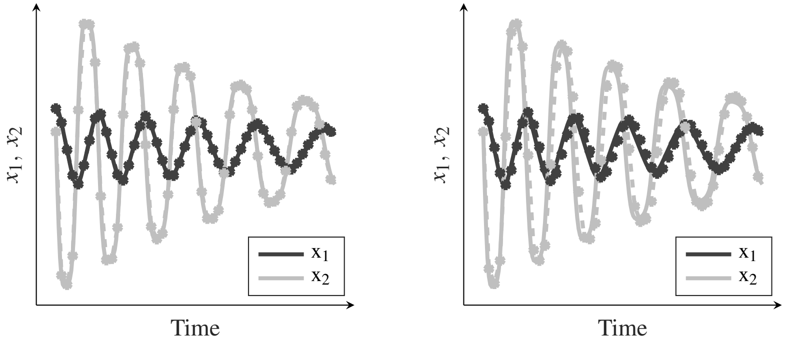
(d) SINDy, noise level $= 10^{-3}$

Figure 4: Test data (solid) and simulated trajectories (star-dash) for the Lotka-Volterra models, that where estimated using I-SINDy (a and b) and SINDy (c and d). Figure (a) is representative of the model accuracy with the noise levels 10^{-4}, 10^{-3}, and 10^{-2}, as there was no significant visible difference between the corresponding model trajectories. Figure (b) shows the model accuracy with a noise level of 10^{-1}. Figures (c) and (d) show the model accuracy for the noise levels 10^{-4} and 10^{-3}, respectively.



(a) I-SINDy, noise level $= 10^{-4}$

(b) I-SINDy, noise level $= 10^{-2}$



(c) SINDy, noise level $= 10^{-3}$

(d) SINDy, noise level $= 10^{-3}$

Figure 5: Training data (solid) and simulated trajectories (star-dash) for the Duffing models that where estimated using I-SINDy (a and b) and SINDy (c and d). The training data is shown because the models performed worse compared against the training data than against the test data (see tables 3 through 6). Figure (a) is representative of the model accuracy with the noise levels 10^{-4} and 10^{-3}, as there was no visible difference between these model trajectories. Figure (b) shows the model accuracy with a noise level of 10^{-2}. Figure (c) is representative of the model accuracy with the noise levels 10^{-4} and 10^{-3}. Figure (d) shows the model accuracy for the noise level 10^{-2}.

Evolving Fuzzy Model Predictive Control based on Optimization for Nonlinear Plate Heat Exchanger

Miha Ožbot, Igor Škrjanc

University of Ljubljana, Faculty of Electrical Engineering
Tržaška 25, 1000 Ljubljana, Slovenia
Email: {miha.ozbot, igor.skrjanc}@fe.uni-lj.si

Summary

In this paper, a novel evolving Fuzzy Model Predictive Control based on Optimization (eFMPC) is proposed. The controller adresses model adaptation in case of abrupt concept shifts in the system parameters. The model of the system is evolved during control based on online learning approaches with a self-monitoring approach to determine the quality of the local models. The control action is a result of optimization based on the Particle Swarm Optimization method. The control principle was tested on the real Plate Heat Exchanger pilot plant.

1 Introduction

Evolving Intelligent Controllers (EIC) can adjust their structure and parameters online in a recursive manner based on the latest available data, making them well-suited for real-time applications with changing system characteristics. This allows the system to condense data into nonlinear dynamic structures while maintaining transparency and interpretability [35], which is of great importance in an industrial setting where results must be reproducible and verifiable. Evolving systems are defined by their ability to *add new* rules when new data becomes available that is not adequately represented by the existing structure, especially in the case of concept drift [41]. Additionally, such systems must be able to *remove* redundant or faulty rules, *merge* clusters to simplify the structure when samples coalesce [17, 55], and perform *splitting* of rules when new concepts appear that conflict with the existing rulebase [40, 41]. Some of the most influential evolving fuzzy and fuzzy systems of the last decade include: eTS+ [7], FLEXFIS+ [37], AnYa [8], FBeM [34], PANFIS [47], eFuMo [19], GS-EFS [38], eGauss+ [51], and others. These have been applied to a variety of distinct problems, showcasing the adaptability of this framework, e.g., data clustering [6, 29, 51], classification [20, 27, 28], nonlinear system identification [13, 22, 39, 48], system control [1, 10], fault detection and diagnostics [3, 5, 12, 19, 24], design of experiments [45, 50], localization and mapping [33], data streaming [16,23,26,41,53], cybersecurity [30,54,56], image segmentation [43, 44], federated learning [46], time-series forecast [18, 21], among others.

Adaptive and predictive control are well established methodologies in the field of control systems engineering that address the limitations of traditional controllers by adapting to changes in the system or predicting future system behavior. These methods rely on an initial model that adapts its parameters over time. An adaptive Predictive Functional Controller (PFC) for hybrid continuous and discrete signals, based on the recursive least squares parameter identification method with exponential forgetting, is presented in [32]. This approach was examined on an exothermic batch reactor, a time-invariant multivariable process. The exponential forgetting factor can lead to an estimator windup phenomenon when persistent excitation is not available, serving as a safety mechanism. Here, the recursive least square gain denominator is used as a measure of proper excitation. A predictive controller with recursive parameter identification for a time-variant linear system is presented in [14]. This method is demonstrated on an unstable third-order system and decomposes the system into an affine linear part and a bias term, which can be quickly adapted when a parameter shift occurs. This approach requires an iterative method to estimate the optimal bias compensation at every time step. An adaptive fuzzy model predictive control (AFMPC) using ant-colony optimization (ACO) is presented in [11]. Fuzzy Takagi-Sugeno type models describe the nonlinear dynamical and static properties of the controlled system. In this approach, recursive least squares parameter identification is halted when the error between the actual and estimated output becomes smaller than a threshold. A model reference adaptive control (MRAC) and a controller output error method (COEM) were used in [58] to control a heating and cooling system with a fuzzy inverse plant model. Interval models are used in [31] to model uncertainties for tuning the parameters of a PID controller with particle swarm optimization (PSO). Closed-loop stability is ensured if the interval-based constraints are not violated. The method was examined on a three-tank hydraulic system and a batch reactor. An unknown nonlinear system was controlled with a data-driven Model-based Predictive Control (MPC) in [9], where future trajectories were computed based on a persistent excitation lemma [57].

However, model-based (indirect or direct) controllers are more challenging to implement because they require an initial model. Evolving systems can identify and adapt a model over time by changing the model's structure in addition to adapting its parameters. Several control algorithms based on evolving systems have been proposed in the literature, with a popular choice for the base model being the evolving Takagi-Sugeno type, as used in the adaptive controller with leakage in the control law [10], the two degrees-of-freedom control with feedforward and feedback components [59], self-tuning predictive control [60], and evolving PID control [15]. Most similar to our approach is the Robust Evolving Adaptive Controller (ReCCo) proposed in [1, 2], which is based on the AnYa [8] evolving system with a low number of a priori parameters and several safety mechanisms but lacks a splitting and merging mechanism for the evolving law. This controller starts from scratch without any initial fuzzy rules. However, the aforementioned approaches lack cluster merging and removal mechanisms, which are crucial for handling drifting data, as outdated rules can become inaccurate and detrimental to control performance. In addition, our method uses self-monitoring to detect anomalies in the system and remove erroneous clusters.

In this study, we examine an evolving fuzzy predictive functional control with fault detection of a nonlinear dynamical system in a changing environment. The proposed EIC methodology is illustrated in Fig. 1. The main contributions of the proposed eFMPC are:

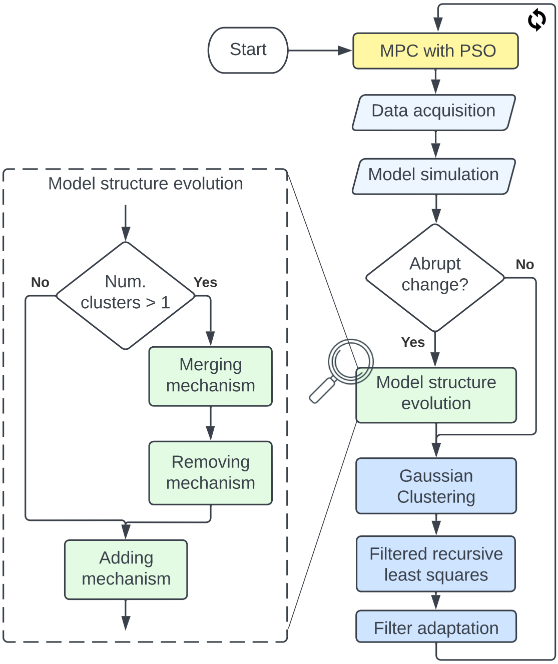


Figure 1: Software flow diagram of the proposed evolving intelligent control.

- An evolving fuzzy model identification method that can identify a model online during predictive functional control of a nonlinear dynamical system. The algorithm uses evolving mechanisms like rule addition, merging, and removal.
- Detection and compensation of abrupt system shifts with a self-monitoring approach, including recursive computation of the Fuzzy Mean Square Error (fMSE).

In this study, we assume that the controlled system is stable, the model operates in discrete-time and is nonlinear, and the disturbance is additive. The reference signal is limited to step changes, and the system experiences both slow drifting and abrupt shifts in operating conditions. The time horizon is small enough that the system can be approximated with a linear model. The evolving fuzzy system used in this study is based on [4, 45] with two notable additions: the Bhattacharyya overlapping measure and the self-monitoring rule removal mechanism. This method was chosen because it is specifically designed for online identification with single-step signal excitation.

2 Evolving fuzzy model identification

Evolving Neuro-Fuzzy Inference Systems (ENFIS) incorporate *fuzzy logic* within a network structure [55]. They consist of sub-models in the form of Takagi-Sugeno fuzzy rules $\mathcal{R}_i$, where $i = 1, 2, \cdots, c$. Each rule is characterized by a membership function defined by a multivariate Gaussian cluster in the antecedent and a local linear model (LLM) in the consequence. The multivariate Gaussian clusters are chosen to represent the input variables in the antecedent structure due to their demonstrated universal approximation capability and their ability to describe correlations between variables [42, 51]. The membership function $\mu_i(k) \in (\varepsilon, 1]$ for a clustering sample $\underline{z}(k) \in \mathbb{R}^{n_z}$ at time step $k = 0, 1, 2, \ldots$ is computed as

$$\mu_i(k) = \exp\Big(-(\underline{z}(k) - \underline{v}_i)^\top \underline{\Sigma}_i^{-1} (\underline{z}(k) - \underline{v}_i)\Big) + \varepsilon, \tag{1}$$

where $\underline{v}_i \in \mathbb{R}^{n_z}$ and $\underline{\Sigma}_i \in \mathbb{R}^{n_z \times n_z}$ represent the center and covariance matrix of the cluster, respectively, and $\varepsilon \in \mathbb{R}$ is a very small constant ensuring sufficient support throughout the input space.

The output of the evolving fuzzy model is calculated as a weighted sum of the outputs $\hat{y}_i(k) \in \mathbb{R}$ of all fuzzy rules $\mathcal{R}_i$ based on the normalized activation of each membership function $\Psi_i(k) \in (0, 1]$:

$$\hat{y}(k) = \sum_{i=1}^{c} \Psi_i(k) \hat{y}_i(k) = \frac{\sum_{i=1}^{c} \mu_i(k) \underline{\varphi}^\top(k) \underline{\theta}_i}{\sum_{i=1}^{c} \mu_i(k)}, \tag{2}$$

where $\hat{y}_i(k) = \underline{\varphi}^\top(k)\underline{\theta}_i$ is the consequence LLM of the rule $\mathcal{R}_i$, and $\underline{\varphi}(k)^\top = [u(k-1), \cdots, u(k-m), y(k-1), \cdots, y(k-n), 1] \in \mathbb{R}^{n+m+1}$ serves as the shared regressor.

The evolving fuzzy model identification uses a clustering method and a parameter identification method independently with different input signals. During the identification process, each training sample is used solely to identify the most recently added fuzzy rule, which enhances robustness and reduces computational complexity by eliminating the need to compute membership functions at every time step. The incremental multivariate Gaussian clustering is based on Welford's online algorithm for calculating variance [51]:

$$\underline{\nu}_i(n_i+1) = \underline{\nu}_i(n_i) + \frac{1}{n_i+1}\underline{e}_i(k), \tag{3}$$

$$\underline{S}_i(n_i+1) = \underline{S}_i(n_i) + \underline{e}_i(k)\left(\underline{z}(k) - \underline{\nu}_i(n_i+1)\right)^\top, \tag{4}$$

where the clustering error is computed as $\underline{e}_i(k) = \underline{z}(k) - \underline{\nu}_i(n_i)$, and the number of samples belonging to the i-th cluster is incremented as $n_i = n_i + 1$. The normalized covariance matrix is computed as $\underline{\Sigma}_i = \underline{S}_i/n_i$ only when it is required.

Evolving fuzzy systems commonly use the Fuzzily-Weighted Recursive Least Squares (wRLS) optimization method for the identification of the parameter of the consequence models [1,36,55]. However, the error formulation of this method results in the ARX model formulation, which is defined as $A(q)y(k) = B(q)u(k) + r$, while we would prefer an OE model $y(k) = \frac{B(q)}{A(q)}u(k) + r$. This is because the ARX model assumes a colored noise at the output of the system due to the auo-regression of the system output, while the OE model works in parallel to the system and assumes a more realistic white output noise. The OE model can be identified by filtering the regression signals with the denominator of the transfer function of the LLM as $\hat{A}(q)\underline{\varphi}_f(k) = \lim_{q\to 1}\hat{A}(q)\underline{\varphi}(k)$ [45]. The Filtered Recursive Least Squares method (fRLS) is then computed as [52]

$$\underline{\theta}_c(k) = \underline{\theta}_c(k-1) + \underline{\gamma}(k)\left(y_f(k) - \varphi_f^\top(k)\theta_c(k-1)\right), \tag{5}$$

$$\underline{\gamma}(k) = \frac{1}{\underline{\varphi}_f^\top(k)\underline{P}(k-1)\underline{\varphi}_f(k)+1}\underline{P}(k-1)\underline{\varphi}_f(k), \tag{6}$$

$$\underline{P}(k) = \left(\underline{I} - \underline{\gamma}(k)\underline{\varphi}_f^\top(k)\right)\underline{P}(k-1), \tag{7}$$

where $\underline{P}(k)$ is an information matrix. The filter is updated online when enough data is collected, i.e. when the confidence interval of the identified model falls under a threshold value [45].

2.1 Rule addition mechanism

A new rule is added when the control error $e(k) = w(k) - y(k) \in \mathbb{R}$ changes abruptly $|e(k) - e(k-1)| > \kappa_e$. With this approach, a new rule is added when the reference signal changes, a large disturbance occurs, or the system experiences a shift in characteristics. The rule base is updated with $c = c + 1$ and initialized with a new antecedent multivariate Gaussian cluster:

$$n_c = 1, \quad \underline{\nu}_c = \underline{z}(k), \quad \underline{\Sigma}_c = \underline{0}, \tag{8}$$

and a new consequence OE-LLM as:

$$\underline{\theta}_c = \textstyle\sum_{i=1}^{c-1} \Psi_i(k)\underline{\theta}_i, \quad \underline{P}(k) = \underline{P}_0, \tag{9}$$

where $\underline{P}_0 = \alpha_P \mathrm{I} \in \mathbb{R}^{n_\varphi \times n_\varphi}$ is the initial information matrix, with a large constant $\alpha_P \in [10^3, 10^7]$, used to ensure fast parameter convergence. The newly added rule is immediately used in the control law to enable a quick reaction to changes in the process. This requires setting the initial parameters of the rule's consequent linear model to the fuzzily weighted average of the existing rules at the time of creation.

2.2 Rule merging mechanism

Due to drifting data, a merging mechanism is required in cases of overlapping clusters, which can cause irregularities in rule activation and differences in the consequent OE-LLM. Overlapping clusters are detected based on the Bhattacharyya distance between two clusters. For multivariate Gaussian clusters, it is defined as [42]:

$$d_{pq}^{B} = \frac{1}{8}\left(\underline{\nu}_p - \underline{\nu}_q\right)^{\top} \underline{\Sigma}_{pq}^{-1}\left(\underline{\nu}_p - \underline{\nu}_q\right) + \frac{1}{2}\ln\left(\frac{\det(\underline{\Sigma}_{pq})}{\sqrt{\det(\underline{\Sigma}_p)\det(\underline{\Sigma}_q)}}\right), \qquad (10)$$

where $\underline{\Sigma}_{pq} = \frac{1}{2}(\underline{\Sigma}_p + \underline{\Sigma}_q)$ is the average covariance matrix of the two clusters. The consequent OE-LLMs are compared based on the similarity of their transfer functions, which is simplified to the comparison of the steady-state variables [45]:

$$d_{pq}^{K} = \left|\frac{\hat{B}_p(1)}{\hat{A}_p(1)} - \frac{\hat{B}_q(1)}{\hat{A}_q(1)}\right|, \quad d_{pq}^{N} = \left|\frac{\hat{r}_p(1)}{\hat{A}_p(1)} - \frac{\hat{r}_q(1)}{\hat{A}_q(1)}\right|, \qquad (11)$$

where d_{pq}^{K} and d_{pq}^{N} respectively represent the dissimilarity in steady-state gain and bias between the two compared systems.

The rules $\mathcal{R}_p$ and $\mathcal{R}_q$ are merged if the conditions for antecedent proximity and consequence similarity ($d_{pq}^{B} < \kappa_B$, $d_{pq}^{K} < \kappa_K$, and $d_{pq}^{N} < \kappa_r$) are satisfied. The antecedent clusters are merged based on the method proposed in [51], and the consequent parameters $\underline{\theta}_{pq}$ are merged as:

$$\underline{\theta}_{pq}(k) = \frac{n_p(k)\underline{\theta}_p(k) + n_q(k)\underline{\theta}_q(k)}{n_{pq}(k)}, \qquad (12)$$

where $n_{pq}(k) = n_p(k) + n_q(k)$.

2.3 Rule removal mechanism

The proposed evolving system employs self-monitoring to determine the validity of the model online using the interleaved test-then-train or prequential approach [25], meaning that the samples used for model identification are first used to validate the existing model. Since the RLS method aims to minimize the MSE of the output, we use a fuzzy MSE (fMSE) to determine the quality of the local models online. It is computed recursively based on the fuzzy error proposed in [49] as:

$$\mathrm{MSE}_j(k) = \frac{M_j(k-1)\mathrm{MSE}_j(k-1) + \mathrm{e}_j^2(k)}{M_j(k)}, \tag{13}$$

$$M_j(k) = M_j(k-1) + \Psi_j^2(k), \tag{14}$$

where $\mathrm{e}_j(k) = \Psi_j(k)\big(y(k) - \hat{y}_j(k)\big)$ is the fuzzy error of the jth rule. The estimation is initialized with the creation of the rule as $\mathrm{MSE}_j = 0$ and $M_j = 1$. The fuzzy MSE is only computed if the rule membership activation is large enough, as this indicates that the sample should be represented by the rule: $\mu_i(k) > \kappa_\mu$. For samples that are far from every cluster, it is better not to attribute the error to any clusters, as we are not interested in samples that are distant from the rule base and they might even be detrimental to the model accuracy. Rules are removed if they have low accuracy ($\mathrm{MSE}_j > \kappa_{\mathrm{MSE}}$) or are overlapping ($d_{pq}^B > \kappa_B$) but p and q could not be merged, in which case the rule with the higher error is removed.

3 Model-based predictive control based on Predictive functional control concept

In general, a Model-based Predictive Controller (MPC) is a type of controller that uses a model of the controlled process ($y_m(k)$) to optimize a specified criterion function in order to obtain the finite-horizon control law. The criterion function J is generally given as:

$$J = \sum_{i=N_1}^{N_2} \Big(y_m(k+i) - y_r(k+i)\Big)^2 + \lambda \sum_{i=0}^{N_u} \Big(\Delta u(k+i)\Big)^2$$

where $y_m(k+i)$ represents the prediction of the process model output, $y_r(k+i)$ represents the prediction of the reference model, N_1 and N_2 are the lower and upper prediction horizons, $\Delta u(k+i)$ is the change in the control signal in the future, N_u is the control horizon (i.e., the number of future time instants where the control signal is taken into account), and λ is the weighting factor.

The solution in the form of a control law can be obtained analytically if the variables in the function are not subject to constraints. However, when dealing with constraints, the optimal solution should be obtained through optimization as follows:

$$\begin{aligned}
&\min_{\Delta u} J \qquad (15)\\
&\text{subject to:}\\
&u_{\min} \leq u(k+i) \leq u_{\max}, \quad i = 1,\ldots,N_u\\
&\Delta u_{\min} \leq \Delta u(k+i) \leq \Delta u_{\max}, \quad i = 1,\ldots,N_u\\
&y_{\min} \leq y_m(k+i) \leq y_{\max}, \quad i = 1,\ldots,N_2
\end{aligned}$$

3.1 Predictive functional control concept

The concept of predictive functional control is based on the rule of equilibrium, given by the following equation:

$$y_r(k+h) - y_p(k) = y_m(k+h) - y_m(k) \qquad (16)$$

where h stands for the coincidence horizon. This means that the change between the current value of the process output and the predicted reference model at the coincidence horizon should be equal to the difference between the current model output and the predicted model output at the coincidence horizon. The left side of the equation is called the process increment $\Delta_p(h)$, and the right side is called the model increment $\Delta_m(h)$. By comparing the change in the output of the identified model with the change in the output of the reference model, an integrating controller is achieved that can compensate for steady-state bias in the system model.

This means that at every time step k, a prediction of future model and reference model outputs ($y_m(k+i)$, $y_r(k+i)$, $i = 1,..,h$) is computed. This requires a dynamical model of the process, and some assumptions to easily calculate the predictions. We assume that the system is linear time-invariant during the prediction horizon, the reference signal is assumed to be constant $w(k)$, $i = 1,\cdots,h$, it is assumed to deal with mean level control law, which assumes a constant, the mean control signal, $u(k)$, $i = 1, \cdots, h$, and the output of the reference model at time step $y_r(k)$ is set to the measured output of the system $y_p(k)$. The output of a first order reference model at a time step k+h is defined as [32]

$$y_r(k+h) = a_r^h y(k) + (1-a_r^h)w(k), \tag{17}$$

where a_r, $0 < a_r < 1$ stands for the reference model pole in the discrete domain, which is selected by the control design. The reference model should have a faster time constants that the uncontrolled process and its order should be smaller or equal to the order of the system [1].

Taking into account that $y_p(k) = y_r(k)$, the control error is defiend as $e(k) = w(k) - y_r(k)$, and the prediction of control error at the coincidence horizon, with the assumption of constant reference $w(k)$ the following condition for the error is obtained

$$e(k+h) = a_r^h e(k). \tag{18}$$

which describes the exponentially decreasing control error. This implies that the process difference $\Delta_p(h)$, where $w(k) = w(k+h)$, and $y_r(k+h) = w(k) - a_r^h e(k)$, becomes equal

$$\Delta_p(h) = y_r(k+h) - y_p(k) = \left(1 - a_r^h\right) e(k) \tag{19}$$

From the basic rule of equilibrium given in Eq. 16 ($\Delta_m(h) = \Delta_p(h)$), the required value for the predicted model output is obtained. With this value of predicted model, the required dynamic of the whole closed-loop control system will be obtained. This is given as follows

$$y_m(k+h) = y_m(k) + \left(1 - a_r^h\right) e(k) \tag{20}$$

The dynamic of the controlled signal can be defined with the pole of the closed-loop system, and can be limited as $a_r \to 0$, to get the fastest possible response, i.e.

$$y_m(k+h) = y_m(k) + e(k) \tag{21}$$

3.2 Particle Swarm Optimization to obtain control value

The prediction of the model output at the coincidence horizon, $y_m(k+h)$, is now denoted as $y_{m,k+h}(u_k)$, indicating that it depends on the control signal applied at time instant k. The required value for the predicted model output at the coincidence horizon, given in Eq. 20, is now written as $y_{m,k+h}^{goal}$. The Particle Swarm Optimization (PSO) method is used to compute the control action, i.e., u_k or $u(k)$, to find the solution for $y_{m,k+h}(u_k)$ that is closest to $y_{m,k+h}^{goal}$. PSO is used here because it is easy to implement and flexible enough to add additional constraints to the control law if needed [11]. The idea is to evaluate a number of different values, i.e., particles u_k^i, as possible solutions, and to find the one that is optimal in the sense of the following simple criterion function:

$$\min_{u_k^i} J = \min_{u_k^i} \left(y_{m,k+h}^{goal} - y_{m,k+h}(u_k^i) \right)^2$$

If the process variables have constraints, i.e., $u_{min} \leq u_k \leq u_{max}$ and $\Delta u_{min} \leq \Delta u_k \leq \Delta u_{max}$, then the criterion function should be extended as follows:

$$\begin{aligned} \min_{u_k^i} J = \min_{u_k^i} & \left(y_{m,k+h}^{goal} - y_{m,k+h}(u_k^i) \right)^2 + \\ & +\lambda_{cons}\Big(1 + \operatorname{sign}(u_k^i - u_{max})\Big) + \\ & +\lambda_{cons}\Big(1 + \operatorname{sign}(u_{min} - u_k^i)\Big) + \\ & +\lambda_{cons}\Big(1 + \operatorname{sign}(u_k^i - \Delta u)\Big) \end{aligned}$$

where $\lambda_{cons} \gg 1$, to heavily penalize solutions that violate the constraints.

The complete controller is presented in Algorithm 1. In order to adapt the model online the identification must be reliable and have a low number of problem-specific parameters.

Algorithm 1.: Evolving Neuro-Fuzzy Model-based Predictive Control with Particle Swarm Optimization

```
**Input:** $u_{\min}, u_{\max}, \underline{P}_0, \kappa_\mu, \kappa_B, \kappa_K, \kappa_N, \kappa_f, \kappa_e, \kappa_{\mathrm{MSE}}$
**Initialize:**
    $\underline{z} \leftarrow$ initial value, $c \leftarrow 1, \underline{\mu}_c \leftarrow z(0), \underline{S}_c \leftarrow \underline{0}, n_c \leftarrow 1, P \leftarrow \underline{P}_0, \underline{\theta}_c \leftarrow \underline{0}, A_f(z) \leftarrow 1$

**repeat**
    $k \leftarrow k+1$
    Perform MPC with PSO
    Perform measurement and signal filtration
    Compute regression vector $\underline{\varphi}_f(k)$ and clustering vector $\underline{z}(k)$
    Compute model error (Eqs. (1) and (2))
    **if** $|e(k)-e(k-1)| > \kappa_e$ **then**
        **repeat**
            **for** $p = 1$ to $c$ **do**
                Compute measure (Eqs. (10) and (11)) for $q \leftarrow \operatorname{argmin}(d^B_{pq})$
            Find $\operatorname{argmin}_{p,q}(d^B_{pq})$ subject to $(d^K_{pq} < \kappa_K) \wedge (d^N_{pq} < \kappa_N) \wedge (d^B_{pq} < \kappa_B)$
            **if** $(d^B_{pq} < \kappa_B)$ **then**
                Perform rule merging mechanism (Eq. (12))
        **until** no rules can be merged
        Perform rule removal mechanism (Eqs. (13) and (14))
        Perform rule addition mechanism (Eqs. (8) and (9))
    Perform incremental clustering (Eqs. (3) and (4))
    Perform recursive parameter identification (Eqs. (5), (6), and (7))
    Adapt filter $A_f(z)$
**until** end of control
```

4 Simulation study

In this study, we examined the proposed control algorithm on the control problem of a Plate Heat Exchanger (PHE). In the experiment, the control valve of the inlet cold water was abruptly closed, resulting in a shift in the parameters of the system. This allowed us to evaluate the capability of the evolving system to quickly adapt to a shift in parameters and assess the robustness of the proposed controller. The experiment on the PHE was conducted using a theoretical model of a plate heat exchanger as defined in [45], with the same parameters.

The examined theoretical system was sampled with a sampling time of $t_s = 4s$ and had a dead time of $t_d = 4t_s$. A heteroscedastic white Gaussian noise

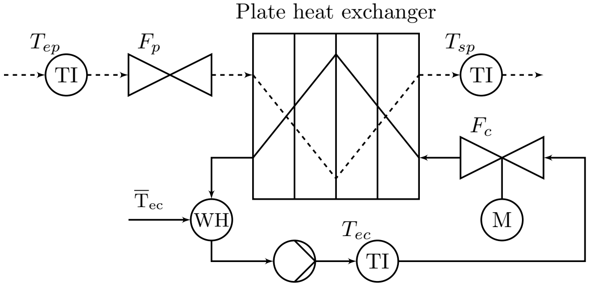


Figure 2: A schematic of the PHE pilot plant. The solid line represents the primary hot water flow circuit, while the dashed line represents the secondary cold water circuit.

$v \sim \mathcal{N}(0, 0.04K)$ was added as output error, where $K \in \mathbb{R}$, is the steady-state gain of the system in the observed operating point. The antecedent clustering vector was selected as $\underline{z}^\top(k) = [u(k), y(k)]$ and the consequence regression vector was selected as $\underline{\varphi}^\top(k) = [u(k-4-1), x(k-4-1), y(k-1), 1]$.

The experiment began with an initial model consisting of 5 rules that were identified using the proposed evolving methodology and a staircase excitation signal. The reference signal was changed every $k_h = 300$ samples to values in the range $w \in [8, 48]$, selected to ensure that the input signal does not reach saturation when the output reaches the reference value. The step height of 10 formed a staircase signal. At the beginning of the seventh step $k{=}6k_h$ the cold water valve F_p was partially closed from a value of 0.53 to 0.4 to simulate a shift in the environment, i.e., a fault of the inlet flow rate.

5 Real PHE Pilot plant study

The proposed eFMPC was used to control a nonlinear dynamical system in a changing environment. The plate heat exchanger pilot plant is subject to changes in the temperature of the inlet cold water and room air throughout the day [1]. The evolving model used for the PFC was built from scratch, with the system creating the first rule based on the first recorded sample without any prior knowledge, to examine the plug-and-play aspect of the proposed approach. However, note that even though the model identification starts from scratch with

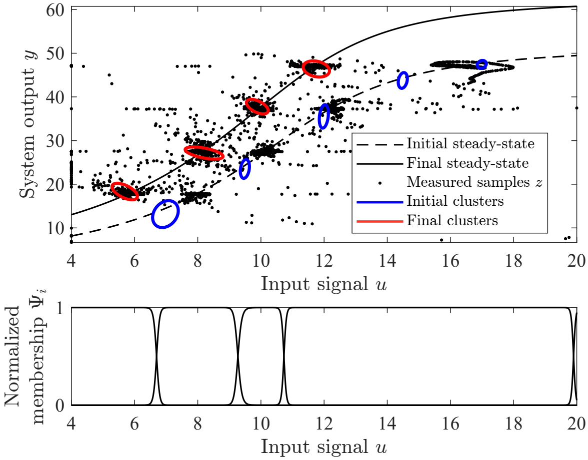


Figure 3: The top figure shows the antecedent multivariate Gaussian clusters at the start (blue ellipsoids) and at the end (red ellipsoids) of the simulation study after the disturbance, with the initial (dashed curve) and final (full curve) steady-state characteristic of the system. The bottom figure shows the final membership functions as a projection of the clusters to the input dimension.

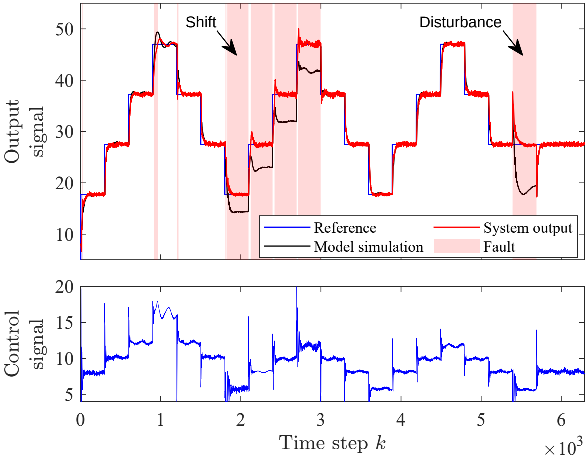


Figure 4: The output of the controlled system, the output of the model, and fault detection (top figure), while the control signal (bottom figure) with the proposed evolving predictive controller.

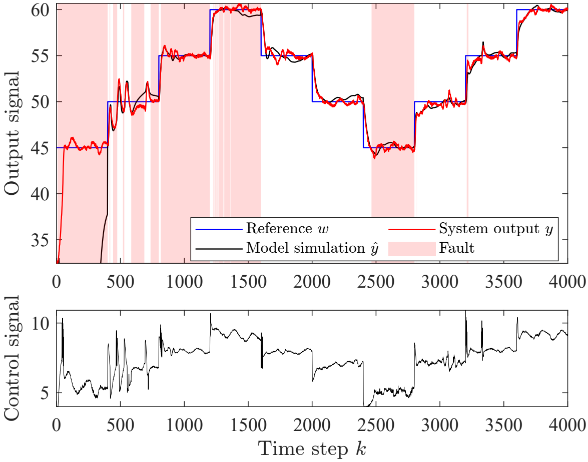


Figure 5: PHE pilot plant control.

a single cluster, the meta-parameters of the algorithm were still selected based on expert knowledge of the system.

6 Discussion

Self-monitoring during online learning present a dilemma in the sense that it is hard to determine whether the error is due to high parameter variance or actual bias. The proposed approach determines the local error of each rule, and only adapts the rule added last in order to avoid creating outlier rules due to large changes in the input signal and to maintain the number of rules as low as possible. This enables the adaptation of the evolving model while detecting low model accuracy, conceptual shifts and disturbances. The phenomena of drifting parameters is commonly addressed with a forgetting factor in the fRLS method, however we omitted it from this implementation as it requires persistent excitation [32] and relied rather on the self-monitoring approach as an alternative that does not suffer from this limitation. Another observation is that the Battacharyya distance works well for detecting overlapping clusters of similar size but can have some difficulties detecting overlapping clusters of different sizes. In general, this can result in small clusters becoming "trapped" inside larger clusters. However, this problem can easily be addressed with removal mechanism based on rule age. For practical application, additional robustness measure could be easily implemented as proposed in [1]. The PSO method is a reliable global optimization method that can find a solution in case of several local minima and it is very flexible to include additional constrains in the control law. However, if the number of particles of the PSO method is not large enough the global minima of the control action might not be found, which results in frequent changes of the input signal.

7 Conclusion

In this study, we proposed an Evolving Fuzzy Model Predictive Controller (eFMPC) that evolves the system model during control. The main benefit is that the model can be used for monitoring, detecting shifts in system parameters, and identifying disturbances as they occur, all while maintaining reference tracking and disturbance regulation. The model was validated through simulations and a real-world study on a plate heat exchanger. A downside of the proposed method is that the evolving identification procedure models every change in the system, including disturbances. Although these are eventually eliminated in the long term, they can affect the controller in the short term. Future work should examine the persistent excitation condition or an information criterion to selectively identify only informative samples.

References

[1] Goran Andonovski, Plamen Angelov, Sašo Blažič, and Igor Škrjanc. A practical implementation of Robust Evolving Cloud-based Controller with normalized data space for heat-exchanger plant. *Applied Soft Computing*, 48:29–38, 11 2016.

[2] Goran Andonovski, Plamen Angelov, Sašo Blažič, and Igor Škrjanc. Robust evolving cloud-based controller (recco). In Igor Škrjanc and Sašo

Blažič, editors, *2017 Evolving and Adaptive Intelligent Systems, EAIS 2017, Ljubljana, Slovenia, May 31 - June 2, 2017*, pages 1–6. IEEE, 2017.

[3] Goran Andonovski, Sašo Blažič, and Igor Škrjanc. Partial cloud-based evolving method for fault detection of HVAC system. In *2018 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, pages 1–6. IEEE, 2018.

[4] Goran Andonovski, Edwin Lughofer, and Igor Škrjanc. Evolving fuzzy model identification of nonlinear wiener-hammerstein processes. *IEEE Access*, 9:158470–158480, 11 2021.

[5] Goran Andonovski, Gasper Mušič, Sašo Blažič, and Igor Škrjanc. Evolving model identification for process monitoring and prediction of non-linear systems. *Eng. Appl. Artif. Intell.*, 68:214–221, 2018.

[6] Goran Andonovski and Igor Škrjanc. Evolving clustering algorithm based on average cluster distance - eCAD. In *IEEE International Conference on Evolving and Adaptive Intelligent System, EAIS 2022, Larnaca, Cyprus, May 25-26, 2022*, pages 1–5. IEEE, 2022.

[7] Plamen Angelov. Evolving Takagi-Sugeno Fuzzy Systems from Streaming Data (eTS+). *Evolving Intelligent Systems: Methodology and Applications*, pages 21–50, 4 2010.

[8] Plamen Angelov and Ronald Yager. A new type of simplified fuzzy rule-based system. *International Journal of General Systems*, 41:163–185, 2012.

[9] Julian Berberich, Johannes Kohler, Matthias A. Muller, and Frank Allgower. Linear Tracking MPC for Nonlinear Systems - Part II: The Data-Driven Case. *IEEE Transactions on Automatic Control*, 67:4406–4421, 9 2022.

[10] Saso Blažič, Igor Škrjanc, and Drago Matko. A robust fuzzy adaptive law for evolving control systems. *Evol. Syst.*, 5(1):3–10, 2014.

[11] S. Bououden, M. Chadli, and H. R. Karimi. An ant colony optimization-based fuzzy predictive control approach for nonlinear processes. *Information Sciences*, 299:143–158, 4 2015.

[12] Oscar Cartagena, Miha Ožbot, Doris Sáez, and Igor Škrjanc. Evolving fuzzy prediction interval for fault detection in a heat exchanger. *Appl. Soft Comput.*, 145:110625, 2023.

[13] Gregor Černe and Igor Škrjanc. Initial study on evolving state space neural networks (essnn). In Plamen Angelov, Yannis Manolopoulos, Edwin Lughofer, and Lazaros S. Iliadis, editors, *2018 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS 2018, Rhodes, Greece, May 25-27, 2018*, pages 1–8. IEEE, 2018.

[14] Gregor Černe and Igor Škrjanc. Cascade affine constant recursive algorithm for model-based control. *IFAC-PapersOnLine*, 54:299–302, 1 2021.

[15] Bruno Sielly Jales Costa, Igor Škrjanc, Sašo Blažič, and Plamen Angelov. A practical implementation of self-evolving cloud-based control of a pilot plant. In *2013 IEEE International Conference on Cybernetics, CYBCO 2013, Lausanne, Switzerland, June 13-15, 2013*, pages 7–12. IEEE, 2013.

[16] Daniel F. Leite, Fernando A. C. Gomide, and Igor Škrjanc. Multiobjective Optimization of Fully Autonomous Evolving Fuzzy Granular Models. In *2019 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2019, New Orleans, LA, USA, June 23-26, 2019*, pages 1–7. IEEE, 2019.

[17] Daniel F. Leite, Igor Škrjanc, and Fernando A. C. Gomide. An overview on evolving systems and learning from stream data. *Evol. Syst.*, 11(2):181–198, 2020.

[18] Dejan Dovžan , Vito Logar, and Igor Škrjanc. Solving the sales prediction problem with fuzzy evolving methods. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2012, Brisbane, Australia, June 10-15, 2012*, pages 1–8. IEEE, 2012.

[19] Dejan Dovžan, Vito Logar, and Igor Škrjanc. Implementation of an Evolving Fuzzy Model (eFuMo) in a Monitoring System for a Waste-Water

Treatment Process. *IEEE Transactions on Fuzzy Systems*, 23:1761–1776, 10 2015.

[20] Dejan Dovžan and Igor Škrjanc. Possible use of evolving c-regression clustering for energy consumption profiles classification. In *2015 IEEE International Conference on Evolving and Adaptive Intelligent Systems, EAIS 2015, Douai, France, December 1-3, 2015*, pages 1–6. IEEE, 2015.

[21] Dejan Dovžan and Igor Škrjanc. Evolving fuzzy model for short-term prediction of energy consumption profiles. In *2016 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS 2016, Natal, Brazil, May 23-25, 2016*, pages 98–102. IEEE, 2016.

[22] Edwin Lughofer and Igor Škrjanc. Evolving Error Feedback Fuzzy Model for Improved Robustness Under Measurement Noise. *IEEE Trans. Fuzzy Syst.*, 31(3):997–1008, 2023.

[23] Edwin Lughofer and Mahardhika Pratama and Igor Škrjanc. Online bagging of evolving fuzzy systems. *Inf. Sci.*, 570:16–33, 2021.

[24] Fabricio P. Lucas, Pyramo Costa, Rose M. S. Batalha, Daniel F. Leite, and Igor Škrjanc. Fault detection in smart grids with time-varying distributed generation using wavelet energy and evolving neural networks. *Evol. Syst.*, 11(2):165–180, 2020.

[25] Dongjiao Ge and Xiao Jun Zeng. Learning data streams online — An evolving fuzzy system approach with self-learning/adaptive thresholds. *Information Sciences*, 507:172–184, 1 2020.

[26] Goran Andonovski, Oscar Sipele, José Antonio Iglesias, Araceli Sanchis, Edwin Lughofer, and Igor Škrjanc. Detection of driver maneuvers using evolving fuzzy cloud-based system. In *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020, Canberra, Australia, December 1-4, 2020*, pages 700–706. IEEE, 2020.

[27] Igor Škrjanc, Goran Andonovski, José Antonio Iglesias, María Paz Sesmero, and Araceli Sanchis. Evolving Gaussian on-line clustering in social network analysis. *Expert Syst. Appl.*, 207:117881, 2022.

[28] Igor Škrjanc. eGauss+ evolving clustering in classification. In *15th IEEE International Symposium on Applied Computational Intelligence and Informatics, SACI 2021, Timisoara, Romania, May 19-21, 2021*, pages 533–538. IEEE, 2021.

[29] Igor Škrjanc. Evolving Cauchy Possibilistic Clustering for Data Stream Modeling and Classification. In *IEEE AFRICON 2023, Nairobi, Kenya, September 20-22, 2023*, pages 1–6. IEEE, 2023.

[30] Igor Škrjanc, Seiichi Ozawa, Tao Ban, and Dejan Dovžan. Large-scale cyber attacks monitoring using Evolving Cauchy Possibilistic Clustering. *Appl. Soft Comput.*, 62:592–601, 2018.

[31] Gorazd Karer and Igor Škrjanc. Interval-model-based global optimization framework for robust stability and performance of pid controllers. *Applied Soft Computing*, 40:526–543, 3 2016.

[32] Gorazd Karer, Igor Škrjanc, and Borut Zupančič. Self-adaptive predictive functional control of the temperature in an exothermic batch reactor. *Chemical Engineering and Processing*, 47:2379–2385, 2008.

[33] Gregor Klančar and Igor Škrjanc. Evolving principal component clustering with a low run-time complexity for LRF data mapping. *Applied Soft Computing*, 35:349–358, 10 2015.

[34] Daniel Leite, Rosangela Ballini, Pyramo Costa, and Fernando Gomide. Evolving fuzzy granular modeling from nonstationary fuzzy data streams. *Evolving Systems*, 3:65–79, 6 2012.

[35] Edwin Lughofer. On-line assurance of interpretability criteria in evolving fuzzy systems – Achievements, new concepts and open issues. *Information Sciences*, 251:22–46, 12 2013.

[36] Edwin Lughofer. Improving the robustness of recursive consequent parameters learning in evolving neuro-fuzzy systems. *Information Sciences*, 545:555–574, 2 2021.

[37] Edwin Lughofer, Jean Luc Bouchot, and Ammar Shaker. On-line elimination of local redundancies in evolving fuzzy systems. *Evolving Systems*, 2:165–187, 9 2011.

[38] Edwin Lughofer, Carlos Cernuda, Stefan Kindermann, and Mahardhika Pratama. Generalized smart evolving fuzzy systems. *Evolving Systems*, 6:269–292, 12 2015. Gen-Smart-EFS.

[39] Edwin Lughofer and Mahardhika Pratama. Online Active Learning in Data Stream Regression Using Uncertainty Sampling Based on Evolving Generalized Fuzzy Models. *IEEE Transactions on Fuzzy Systems*, 26:292–309, 2 2018.

[40] Edwin Lughofer, Mahardhika Pratama, and Igor Škrjanc. Incremental rule splitting in generalized evolving fuzzy regression models. In Igor Škrjanc and Sašo Blažič, editors, *2017 Evolving and Adaptive Intelligent Systems, EAIS 2017, Ljubljana, Slovenia, May 31 - June 2, 2017*, pages 1–8. IEEE, 2017.

[41] Edwin Lughofer, Mahardhika Pratama, and Igor Škrjanc. Incremental Rule Splitting in Generalized Evolving Fuzzy Systems for Autonomous Drift Compensation. *IEEE Trans. Fuzzy Syst.*, 26(4):1854–1865, 2018.

[42] Edwin Lughofer and Moamar Sayed-Mouchaweh. Autonomous data stream clustering implementing split-and-merge concepts – Towards a plug-and-play approach. *Information Sciences*, 304:54–79, 5 2015.

[43] Miloš Antić, Andrej Zdešar, and Igor Škrjanc. Depth-Image Segmentation Based on Evolving Principles for 3D Sensing of Structured Indoor Environments. *Sensors*, 21(13):4395, 2021.

[44] Miloš Antić, Andrej Zdešar, and Igor Škrjanc. Segmentation of Stereo-Camera Depth Image into Planar Regions based on Evolving Principal Component Clustering. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI 2021, Karlsruhe, Germany, September 23-25, 2021*, pages 1–5. IEEE, 2021.

[45] Miha Ožbot, Edwin Lughofer, and Igor Škrjanc. Evolving Neuro-Fuzzy Systems based Design of Experiments in Process Identification. *IEEE Transactions on Fuzzy Systems*, pages 1–11, 2022.

[46] Ožbot, Miha and Ozawa, Seiichi and Škrjanc, Igor. eFedGauss: A Federated Approach to Fuzzy Multivariate Gaussian Clustering. In *2024 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, page 1–8, June 2024.

[47] Mahardhika Pratama, Sreenatha G. Anavatti, Plamen P. Angelov, and Edwin Lughofer. PANFIS: A novel incremental learning machine. *IEEE Transactions on Neural Networks and Learning Systems*, 25:55–68, 1 2014.

[48] Sašo Blažič and Igor Škrjanc. Incremental Fuzzy C-Regression Clustering From Streaming Data for Local-Model-Network Identification. *IEEE Trans. Fuzzy Syst.*, 28(4):758–767, 2020.

[49] Igor Škrjanc. Fuzzy confidence interval for pH titration curve. *Applied Mathematical Modelling*, 35:4083–4090, 8 2011.

[50] Igor Škrjanc. Evolving Fuzzy-Model-Based Design of Experiments with Supervised Hierarchical Clustering. *IEEE Transactions on Fuzzy Systems*, 23:861–871, 8 2015.

[51] Igor Škrjanc. Cluster-Volume-Based Merging Approach for Incrementally Evolving Fuzzy Gaussian Clustering-eGAUSS+. *IEEE Transactions on Fuzzy Systems*, 28:2222–2231, 9 2020.

[52] Igor Škrjanc. An evolving concept in the identification of an interval fuzzy model of Wiener-Hammerstein nonlinear dynamic systems. *Information Sciences*, 581:73–87, 12 2021.

[53] Igor Škrjanc, Sašo Blažič, Edwin Lughofer, and Dejan Dovžan. Inner matrix norms in evolving Cauchy possibilistic clustering for classification and regression from data streams. *Inf. Sci.*, 478:540–563, 2019.

[54] Igor Škrjanc, Araceli Sanchis de Miguel, José Antonio Iglesias, Agapito Ledezma, and Dejan Dovžan . Evolving cauchy possibilistic clustering

based on cosine similarity for monitoring cyber systems. In Igor Škrjanc and Sašo Blažič, editors, *2017 Evolving and Adaptive Intelligent Systems, EAIS 2017, Ljubljana, Slovenia, May 31 - June 2, 2017*, pages 1–5. IEEE, 2017.

[55] Igor Škrjanc, Jose Iglesias, Araceli Sanchis, Daniel Leite, Edwin Lughofer, and Fernando Gomide. Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A Survey. *Information Sciences*, 490:344–368, 7 2019.

[56] Igor Škrjanc, Seiichi Ozawa, Dejan Dovžan , Tao Ban, Junji Nakazato, and Jumpei Shimamura. Evolving cauchy possibilistic clustering and its application to large-scale cyberattack monitoring. In *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017, Honolulu, HI, USA, November 27 - Dec. 1, 2017*, pages 1–7. IEEE, 2017.

[57] Jan C. Willems, Paolo Rapisarda, Ivan Markovsky, and Bart L.M. De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54:325–329, 4 2005.

[58] Sofia Yousuf and Muhammad Bilal Kadri. Comparison of two different model free fuzzy control architectures based on inverse plant modeling. *2017 1st International Conference on Latest Trends in Electrical Engineering and Computing Technologies, INTELLECT 2017*, 2018-January:1–7, 1 2018.

[59] Andrej Zdešar, Dejan Dovžan, and Igor Škrjanc. Self-tuning of 2 DOF control based on evolving fuzzy model. *Appl. Soft Comput.*, 19:403–418, 2014.

[60] Žiga Stržinar and Igor Škrjanc. Self-tuned Model-Based Predictive Control Using Evolving Fuzzy Model of a Non-linear Dynamic Process. In *Explainable AI and Other Applications of Fuzzy Techniques - Proceedings of the 2021 Annual Conference of the North American Fuzzy Information Processing Society, NAFIPS 2021, Virtual Event / West Lafayette, IN, USA, June 7-9, 2021*, volume 258 of *Lecture Notes in Networks and Systems*, pages 406–421. Springer, 2021.

Optimized Excitation Signal Design Employing Receding Horizon Control

Max Heinz Herkersdorf[1], Oliver Nelles[2]

University of Siegen
Paul-Bonatz-Straße 9-11, 57076 Siegen
[1]E-Mail: max.herkersdorf@uni-siegen.de
[2] E-Mail: oliver.nelles@uni-siegen.de

Introduction

Contemporary nonlinear system identification applications leverage powerful machine learning techniques to a great extent. The effectiveness of these data-driven approaches is significantly influenced by the quality of the input or excitation signals employed to generate training and validation datasets. Consequently, alongside the selection of an appropriate model architecture and parameter estimation strategy, the methodologies for input signal design are of paramount importance.

The fundamental objective of input signal design methodologies is to obtain precise and comprehensive information concerning the process behaviors intended to be modeled. Especially considering real-world limitations like time and process constraints as well as high measurement costs [1], designing input signals becomes a highly application-specific challenge. This challenge inherently involves a tradeoff between acquiring information in unknown operational areas (exploration) and refining knowledge in established areas (exploitation).

This contribution presents a novel strategy for generating excitation signals tailored to nonlinear dynamic processes. Drawing inspiration from receding horizon control (RHC), the excitation signal is optimized in an iterative manner, with each iteration focusing solely on a finite time horizon. In the optimization process, a newly introduced criterion is pursued, enabling user-defined adaptations of information acquisition to align with relevant process behaviors while also accommodating real-world limitations. This flexibility enables the method to respond effectively to application-specific challenges.

The proposed approach aligns with a recent research field focused on optimizing the distribution within the input space of a nonlinear dynamic process [3–5]. These approaches are grounded in the principle that the information collected by an excitation signal about a process is inherently linked to its generated distribution within the process's input space. Consequently, the challenge of "gathering information about the process behavior intended to be modeled" can be reframed as the task of "exciting the relevant regions in the process's input space". The underlying assumption is that, if all relevant regions of the input space are sufficiently represented in the training (and validation) datasets, a model can be trained that will be able to accurately describe the process behavior of interest. This is particularly applicable to Markovian processes, which can be fully predicted given knowledge of the current process states [6].

Excitation Signal Design Strategy

In this section, the proposed excitation signal design methodology is presented, emphasizing its key contributions: (i) the development of a RHC-like iterative approach, drawing inspiration from [2] and (ii) the introduction of a novel optimization criterion, calculated in the input space of a nonlinear dynamic model and designed to flexibly respond to application-specific challenges.

Receding Horizon Control-Like Algorithm

The main idea of the RHC-like excitation signal design is rooted in its iterative optimization, which is performed only within the finite time horizon L each iteration. As optimized data points $\underline{\hat{U}}_{\mathrm{opt}}$ become available through the optimization (cf. Eq. 1), the optimal data point $\underline{\hat{u}}_{\mathrm{opt}}(k)$ is appended to the existing signal $\underline{U}$. The time horizon is then shifted forward and the optimization process is repeated until the entire signal is designed. Constraining the controllable inputs and the model input space through $\mathcal{U}$ and $\mathcal{X}$, respectively, is highly advantageous for real-world applications [7].

The core aim of the optimization is to generate data points in the relevant regions of the process's input space, thereby facilitating the collection of information about its behavior intended to be modeled. Typically, prior knowledge regarding the process is however limited, rendering direct access to the input space infeasible. Consequently, a surrogate model M_θ is employed that substitutes the process input space distribution $\underline{X}$ with the model input space distribution $\tilde{\underline{X}}$ and thus allows for the calculation of J.

Algorithm 1.: The RHC-Like Optimization Using Simulated Annealing

Parameters: Number of data points N, weight coefficients $\underline{q}$, time horizon L.
Initialization: Constrained space of the controllable inputs $\mathcal{U}$, constrained model input space $\mathcal{X}$, distance metric dataset $\underline{\Psi}$, surrogate model $\mathcal{M}_\theta$, initial model state $\tilde{\underline{x}}(0)$, dimension p of the model's input space.

```
for $k = 1,2,\ldots,N$ do                                          ▷ Timesteps
    $$\begin{aligned}
    &\underline{\hat{U}}_{\text{opt}} = \underset{\underline{\hat{U}}}{\arg\min}\, J\big(\tilde{\underline{X}},\underline{\Psi},\underline{q}\big) \\
    &\text{with } \tilde{\underline{X}} = \mathcal{M}_\theta(\underline{\hat{U}},\underline{U},\tilde{\underline{x}}(0)) \\
    &\text{s.t. } \underline{\hat{u}}(j) \in \mathcal{U} \quad \forall j = \{k,k+1,\ldots,k+L-1\} \\
    &\qquad \underline{u}(j) \in \mathcal{U} \quad \forall j = \{1,2.\ldots,k-1\} \\
    &\qquad \tilde{\underline{x}}(j) \in \mathcal{X} \quad \forall j = \{0,1\ldots,k,\ldots,k+L-1\} \\
    &\qquad \underline{\psi}(j) \in \mathbb{R}^p \quad \forall j = \{1,2,\ldots,N_\Psi\} \\
    &\qquad q(j) \in \mathbb{R} \quad \forall j = \{1,2,\ldots,N_\Psi\}
    \end{aligned} \tag{1}$$
    Apply $\underline{U} \leftarrow \underline{U} \cup \underline{\hat{u}}_{\text{opt}}(k)$.     ▷ Append optimal data point
    Optimize $\mathcal{M}_\theta$ with data from $\underline{U}$.          ▷ Only for active learning approach
    Go to $k = k+1$.
```

Novel Optimization Criterion

The optimization criterion can be mathematically formulated as follows:

$$\begin{aligned} J(\underline{\tilde{X}},\underline{\Psi},\underline{q}) &= \sum_{j=1}^{N_\Psi} q(j)\cdot d_{\mathrm{NN}}\big(\underline{\psi}(j),\underline{\tilde{X}}\big) \\ \text{with}\ \ d_{\mathrm{NN}}\big(\underline{\psi}(j),\underline{\tilde{X}}\big) &= \min_{1\le o\le N_{\underline{\tilde{X}}}} \left|\underline{\tilde{x}}(o)-\underline{\psi}(j)\right| . \end{aligned} \tag{2}$$

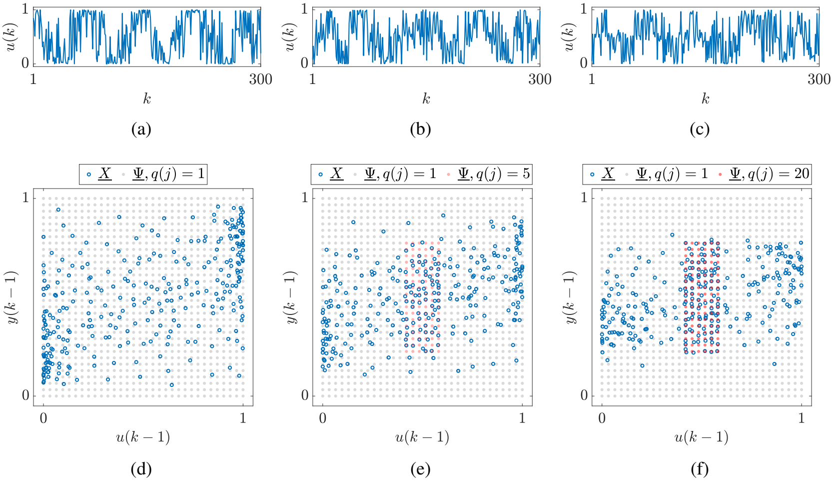
In this formulation, $\underline{\Psi}$ represents a distance metric dataset, uniformly distributed within the region of $\underline{\tilde{X}}$, d_{NN} denotes the nearest neighbor-distance, and $\underline{q}$ contains user-defined weighting coefficients. Hence, J can be interpreted as the weighted sum of the nearest neighbor distances from each point in $\underline{\Psi}$ to $\underline{\tilde{X}}$. By adjusting the values within $\underline{q}$, different regions of the model input space can be emphasized with varying intensities.

It is crucial to recognize that minimizing the discrepancies between $\underline{\tilde{X}}$ and $\underline{X}$ is essential for optimal performance. Encouragingly, Heinz et al. (2017) demonstrated that even a linear time-invariant (LTI) surrogate can produce satisfactory outcomes. However, for processes characterized by substantial nonlinear behaviors that lead to significant divergences between the LTI and the real input spaces, a more sophisticated active learning methodology may be employed, wherein M_θ is continually refined using data derived from $\underline{U}$.

Evaluation

This section sheds light on the effectiveness of the proposed method in adapting to application-specific requirements by concentrating the information acquisition on process behavior intended to be modeled. Specifically, it is demonstrated how adjusting the weighting coefficients enables targeted emphasis on different regions of the process input space, thereby facilitating a flexible balance between exploration and exploitation.

Figure 1 illustrates excitation signals and their corresponding distributions in the process input space, generated using distinct weighting schemes. The test process employed is a nonlinear first-order Hammerstein system. Hence, $\underline{x}^T(k) = [u(k-1), y(k-1)]$ and $\underline{X} = [\underline{x}(1), \underline{x}(2), \ldots, \underline{x}(N)]$ with N data points. A first-order LTI system is used as surrogate model. In Fig. 1 (d), an equal weighting of the nearest neighbor distances to each point in $\underline{\Psi}$ is applied, resulting in a high-quality space-filling design. This approach is suitable when minimal prior knowledge about the process is available, and the goal is to explore unknown operational regions [5]. However, if the application requires an intensified information acquisition in regions of higher interest, this can be achieved by increasing the weights of the nearest-neighbor distances calculated to the points of $\underline{\Psi}$ in these regions. An example of such targeted exploitation is shown in Fig. 1 (e) and (f). The progressively increased weighting, illustrated by the red dots, results in a greater concentration of data points in the region of higher interest. This is accompanied by enhanced information acquisition regarding the process behavior in this area.



(a) (b) (c)

(d) (e) (f)

Figure 1: Excitation signals and below the corresponding process input space distributions employing the example of a nonlinear first-order Hammerstein system. When calculating distances to the red-dotted points of $\underline{\Psi}$, an increased weighting was applied.

Conclusion

A novel excitation signal design strategy based on a receding horizon control-inspired optimization has been presented. The proposed method has been shown to effectively generate space-filling designs within the input space of a nonlinear dynamic process, thereby enabling sophisticated acquisition of information in previously unexplored operational areas. Additionally, the strategy can intensify the exploitation of specific operational areas during information gathering, offering flexibility in meeting application-specific requirements.

References

[1] Rivera, D.E., Lee, H., Mittelmann, H.D., and Braun, M.W. (2009). “Constrained multisine input signals for plant-friendly identification of chemical process systems”. In *Journal of Process Control*, 19(4), 623-635.

[2] Gedlu, E.G., Wallscheid, O., Boecker, J., and Nelles, O. (2023). “Online system identification and excitation for thermal monitoring of electric machines using machine learning and model predictive control”. In *IEEE 14th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)*, 109-115.

[3] Herkersdorf, M.H., Koesters, T., and Nelles, O. (2024). “Optimized Excitation Signal Tailored to Pertinent Dynamic Process Characteristics” [Unpublished Manuscript]. In *IFAC 4th Modeling, Estimation and Control Conference (MECC).*

[4] Kiss, M., Toth, R., and Schoukens, M. (2024). “Space-Filling Input Design for Nonlinear State-Space Identification”. In *arXiv preprint arXiv:2405.18207.*

[5] Heinz, T.O. and Nelles, O. (2017). “Iterative excitation signal design for nonlinear dynamic black-box models”. In *Procedia computer science*, 112, 1054-1061.

[6] Van Kampen, N.G. (1992). “Stochastic Processes in Physics and Chemistry”. North-Holland Publishing.

[7] Koesters, T., Heinz, T.O., and Nelles, O. (2022). “Optimization based excitation signal design tailored to application specific requirements”. In *IFAC-PapersOnLine*, 55(37) 451-456.

Passive-based Analysis and Control of Takagi-Sugeno Fuzzy Systems

Horst Schulte

Control Engineering Group, School of Engineering – Energy and
Information, University of Applied Sciences, HTW Berlin, Germany
Email: schulte@htw-berlin.de

Summary

In this paper, sufficient LMI-based conditions for general Quadratic Supply Rate (QSR) functions, known as QSR-dissipativity of nonlinear systems in Takagi-Sugeno form are proposed. To determine the stability of negative feedback loops of two systems, it is sufficient to prove the passivity of each individual system. This property can be used to ensure the stability of even globally distributed systems if the individual local systems are passive. This is particularly relevant for the stabilization of massively distributed power systems, for example. The passivity can either be a property of non-regulated systems or non-passive systems becomes passive through feedback controller. Both cases are analyzed for the class of Takagi-Sugeno fuzzy systems.

1 Introduction

The starting point of this investigation based on the system-theoretical observation that a negative feedback loop consisting of two passive systems is passive, which is a sufficient condition for the stability of negative feedback loops. This means that networked systems comprise passive elements whose negative feedback loops are successively combined to form a passive system. In contrast, negative feedback loops can lead to instability if the individual systems are only stable. This behavior has been observed since the beginning of mathematical investigation of feedback systems in the frequency and time domain. A central concept for passive systems is the consideration of storage functions. Methods that finds a Lyapunov function to prove stability, can also be used to find a storage function. The relationship between dissipativity as a generic concept and Lyapunov stability can be established by employing the storage function $S(x)$ as a Lyapunov function $V(x)$, where $V(x) \geq 0$ for all $x \in X$ is a positive semidefinite function $\mathbb{R}^n \to \mathbb{R}$

$$V(x(t)) - V(x(0)) \leq \int_0^t f_s(u(\tau), y(\tau))\, d\tau \tag{1}$$

with the mathematical formulation of the supply rate $f_s(u(\tau), y(\tau))$, where $u \in \mathbb{R}^m$ denotes the system input and $y \in \mathbb{R}^p$ the system output. The mathematical formulation (1) is called the dissipation inequality. If the supply rate can be described with the bilinear form $f_s(u,y) = u^T y$, the dissipation inequality is related to the definition of passive systems. According to (1), passivity is, therefore, the property that the increase in storage S described mathematically by a storage function is not larger than the bilinear supply rate. Note $f_s(u,y) = u^T y$ requires that the number of system inputs and outputs must be the same. This indicates the appropriate choice of inputs and outputs is already a part of the system analysis and controller synthesis. However, the dissipation inequality (1) is a stronger criterion than the Lyapunov criterion (related to Lyapunov's second method for stability) [5], [7]) which has previously been used in the Takagi-Sugeno (T-S) framework for numerical determination with LMI constraints for synthesis and analysis [2], [3], and [8].

This paper is organized as follows: Section 2 introduces the definition and formulation of the problem, which are taken from the textbooks [9] and [10]. LMI criteria to verify the dissipation properties of LTI and T-S systems are proposed in Section 3. Section 4 deals with the controller design approach to transform non-passive systems into passive systems through appropriate feedback control. An illustrating example is proposed and discussed in Section 5.

2 QSR-Dissipativity and Passivity of Systems

A special form of the supply function in (1) is the general quadratic supply rate (QSR) also known as QSR-dissipativity defined as

$$f_s(u,y) = \begin{pmatrix} y(t) \\ Tu(t) \end{pmatrix}^T \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \begin{pmatrix} y(t) \\ u(t) \end{pmatrix} . \qquad (2)$$

The choice of the Q-, S- and R-matrices determines the the system category and, with the specified matrix definitions, leads to

- *passive systems* with $Q = 0_{p,p}$, $S = \frac{1}{2}I_p$, and $R = 0_{p,p}$, where $0_{p,p}$ denotes the null matrix and I_p the identity matrix.
- *strictly passive systems* with $Q = -\varepsilon I_p$, $S = \frac{1}{2}I_p$, and $R = -\delta I_p$
  - a distinction is made between *strictly input passive* $\delta > 0$, *strictly output passive* $\varepsilon > 0$, and *very strictly passive* with $\delta > 0$ and $\varepsilon > 0$.
- $\mathcal{L}_2$-*gain systems* with $Q = -I_p$, $S = 0_{p,m}$, and $R = \gamma I_m$

$\mathcal{L}_2$-gain systems meet the norm criterion for the input/output signal given as

$$\underbrace{\int_0^\infty y^T y \, d\tau}_{\|y\|_2^2} \leq \gamma \underbrace{\int_0^\infty u^T u \, d\tau}_{\|u\|_2^2} + V(x_0) , \quad x_0 := x(t_0) . \qquad (3)$$

In the following section, LMI criteria for LTI and T-S systems will be proposed for analysis and passivity-based controller synthesis.

3 Criteria for General Dissipativity Analysis of LTI and T-S Systems

Derivation of the integral form (1) and the substitution of $f_s(u,y)$ by the quadratic supply rate (2) results in

$$\dot{V}(x(t)) \leq \begin{pmatrix} y(t) \\ u(t) \end{pmatrix}^T \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \begin{pmatrix} y(t) \\ u(t) \end{pmatrix}, \tag{4}$$

where $Q = Q^T$ and $R = R^T$. The analysis below is done with the quadratic Lyapunov function candidate

$$V(x(t)) = x^T P x, \quad P \succ 0, \quad P = P^T, \tag{5}$$

$$\dot{V}(x(t)) = \dot{x}^T P x + x^T P \dot{x}. \tag{6}$$

3.1 Dissipativity Analysis of LTI Systems

As a first step, criteria for the dissipativity analysis of LTI state-space systems

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad y(t) = Cx(t) + Du(t) \tag{7}$$

are derived. The substitution of $\dot{x}$ in (6) by the right hand side (rhs) of the state differential equation results in

$$\dot{V}(x) = \big(Ax + Bu\big)^T Px + x^T P\big(Ax + Bu\big). \tag{8}$$

For reasons of simplification, the notation of the time dependency of the variables is omitted. After a brief rearrangement of (8), we obtain

$$\dot{V}(x) = \begin{pmatrix} x \\ u \end{pmatrix}^T \underbrace{\begin{pmatrix} PA + A^T P & PB \\ B^T P & 0 \end{pmatrix}}_{\prec 0} \begin{pmatrix} x \\ u \end{pmatrix} < 0. \tag{9}$$

With (9) as the left hand side of (4), the dissipation inequality in matrix form is obtained

$$\begin{pmatrix} x \\ u \end{pmatrix}^T \begin{pmatrix} PA + A^T P & PB \\ B^T P & 0 \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} \leq \begin{pmatrix} y \\ u \end{pmatrix}^T \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \begin{pmatrix} y \\ u \end{pmatrix} . \tag{10}$$

Using the output equation of (7), the rhs becomes

$$\begin{aligned} \dots \leq & \begin{pmatrix} Cx + Du \\ u \end{pmatrix}^T \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \begin{pmatrix} Cx + Du \\ u \end{pmatrix} \\ &= x^T C^T Q C x + x^T C^T Q D u + x^T C^T S u + u^T S^T C x \\ &\quad + u^T D^T Q C x + u^T S^T D u + u^T D^T S u + u^T R u + u^T D^T Q D u \\ &= \begin{pmatrix} x \\ u \end{pmatrix}^T \begin{pmatrix} C^T Q C & C^T Q D + C^T S \\ D^T Q C + S^T C & D^T Q D + D^T S + S^T D + R \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} . \end{aligned}$$

In combination with the left hand side (lhs) of (10), one obtains

$$\begin{pmatrix} x \\ u \end{pmatrix}^T \underbrace{\begin{pmatrix} PA + A^T P - C^T Q C & PB - C^T Q D - C^T S \\ B^T P - D^T Q C - S^T C & -D^T Q D - D^T S - S^T D - R \end{pmatrix}}_{\preccurlyeq 0} \begin{pmatrix} x \\ u \end{pmatrix} \leq 0 \tag{11}$$

This leads with (5) to the LMI criterion for the general QSR dissipativity of LTI systems in state space form

$$P \succ 0 , \quad \begin{pmatrix} PA + A^T P - C^T Q C & PB - C^T Q D - C^T S \\ B^T P - D^T Q C - S^T C & -D^T Q D - D^T S - S^T D - R \end{pmatrix} \preccurlyeq 0 . \tag{12}$$

Based on the definitions of Q, S, and R in Section 2, the passive, strictly passive and $\mathcal{L}_2$-gain property can be analyzed by solving for a feasible P.

3.2 Dissipativity Analysis of T-S Systems

The results for LTI systems are now to be transferred to T-S systems of the form [8]

$$\dot{x}(t) = \sum_{i=1}^{N_r} h_i(z(t))\big(A_i x(t) + B_i u(t)\big), \quad y(t) = \sum_{i=1}^{N_r} h_i(z(t))\big(C_i x(t) + D_i u(t)\big)\,, \tag{13}$$

where $h_i(z) : \mathbb{R}^l \to \mathbb{R}$ fulfill the convex sum condition

$$\sum_{i=1}^{N_r} h_i(z) = 1, \quad h_i(z) \geq 0\,. \tag{14}$$

The derivation of the quadratic Lyapunov function (6) related to T-S system (13) is given as

$$\dot{V}(x) = \begin{pmatrix} x \\ u \end{pmatrix}^T \sum_{i=1}^{N_r} h_i(z) \begin{pmatrix} PA_i + A_i^T P & PB_i \\ B_i^T P & 0 \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix}. \tag{15}$$

Thus the dissipation inequality (4) for T-S systems is obtained as

$$\sum_{i=1}^{N_r} h_i(z) \begin{pmatrix} x \\ u \end{pmatrix}^T \begin{pmatrix} PA_i + A_i^T P & PB_i \\ B_i^T P & 0 \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} \leq \begin{pmatrix} y \\ u \end{pmatrix}^T \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \begin{pmatrix} y \\ u \end{pmatrix}. \tag{16}$$

Substitution of y in rhs of (16) by the output equation (13) yields

$$\begin{aligned} \dots \leq & \begin{pmatrix} \sum_{i=1}^{N_r} h_i(z)\big(C_i x + D_i u\big) \\ u \end{pmatrix}^T \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \begin{pmatrix} \sum_{j=1}^{N_r} h_j(z)\big(C_j x + D_j u\big) \\ u \end{pmatrix} \\ = & \, x^T \sum_{i=1}^{N_r} \sum_{j=1}^{N_r} h_i(z) h_j(z) C_i^T Q C_j x + x^T \sum_{i=1}^{N_r} \sum_{j=1}^{N_r} h_i(z) h_j(z) C_i^T Q D_j u \\ & + x^T \sum_{i=1}^{N_r} h_i(z) C_i^T S u + u^T S^T \sum_{j=1}^{N_r} h_j(z) C_j x + u^T \sum_{i=1}^{N_r} \sum_{j=1}^{N_r} h_i(z) h_j(z) D_i^T Q C_j x \end{aligned}$$

$$+u^T S^T \sum_{i=1}^{N_r} h_j(z) D_j u + u^T \sum_{i=1}^{N_r} D_i^T S u + u^T R u + u^T \sum_{i=1}^{N_r} \sum_{j=1}^{N_r} D_i^T Q D_j u \,.$$

Utilizing the convex sum condition (14) of $h_i(z)$ resp. $h_j(z)$ the rhs of (16) results in

$$\begin{pmatrix} x \\ u \end{pmatrix}^T \sum_{i=1}^{N_r} \sum_{j=1}^{N_r} h_i(z) h_j(z) \begin{pmatrix} C_i^T Q C_j & C_i^T Q D_j + C_i^T S \\ D_i^T Q C_j + S^T C_j & D_i^T Q D_j + D_i^T S + S^T D_j + R \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix}$$

By using the compact notation proposed in [11], we obtain with the lhs of (16) the inequality

$$\sum_{i=1}^{N_r} \sum_{j=1}^{N_r} h_i(z) h_j(z) \begin{pmatrix} x \\ u \end{pmatrix}^T \Gamma_{ij} \begin{pmatrix} x \\ u \end{pmatrix} \leq 0 \,, \tag{17}$$

where

$$\Gamma_{ij}(P) = \begin{pmatrix} PA_i + A_i^T P - C_i^T Q C_j & PB_i - C_i^T Q D_j - C_i^T S \\ B_i^T P - D_i^T Q C_j - S^T C_j & -D_i^T Q D_j - D_i^T S - S^T D_j - R \end{pmatrix} . \tag{18}$$

With the upper bound, if

$$\begin{pmatrix} x \\ u \end{pmatrix}^T \Gamma_{ij}(P) \begin{pmatrix} x \\ u \end{pmatrix} \leq 0 \tag{19}$$

for each term $i, j = 1, \ldots, N_r$ holds, this also valid for the total sum (17). Finally, utilizing the symmetry of the multiplication of the h_i-functions, the relaxed LMI condition [8] to verify the QSR-dissipativity of T-S system is obtained:

$$\begin{aligned} & P \succ 0 \,, \\ & \Gamma_{ij}(P) + \Gamma_{ji}(P) \preccurlyeq 0 \,, \\ & \Gamma_{ii}(P) \preccurlyeq 0 \quad \text{for all} \quad i = 1, 2, \ldots, N_r \,, \quad j = i+1, i+2, \ldots, N_r \\ & \qquad\qquad s.t. \quad h_i(z) h_j(z) \neq 0, \ \exists z \end{aligned} \tag{20}$$

with $\Gamma_{ij}(P)$ proposed in (18). Based on the definitions of Q, S, and R in Section 2, the passive, strictly passive and $\mathcal{L}_2$-gain properties of nonlinear systems in T-S form (13) can be analyzed by finding a common P. How to apply (20) in principle by solving a convex optimization problem is shown in Section 5 by a toy example. The applicability of (20) to real systems, for example wind and photovoltaic generator models of power systems in T-S form proposed in [11], will be investigated in future studies.

4 Passivity-Based Control of T-S systems

To analyze the passive-based control of a T-S system, the state-space model without direct pass-through is considered:

$$\dot{x} = \sum_{i=1}^{N_r} h_i(z)\big(A_i + B_i u\big)\,, \qquad y = \sum_{i=1}^{N_r} h_i(z)\,C_i x\,. \tag{21}$$

The proposed control law

$$u = \sum_{j=1}^{N_r} h_j(z)\big(K_j x + F_j v\big)\,, \tag{22}$$

consists of a state feedback matrix K_j and gain matrix F_j to feed-forward the reference signal $v \in \mathbb{R}^p$ in order to achieve a steady-state control error of zero. The structure of the control law corresponds to the parallel distributed compensator (PDC) of Takagi-Sugeno fuzzy systems [8], whereby the weighting functions are identical to those of the model equation (21). Obtaining the closed-loop system the input in (21) is substituded by the control law (22). Utilizing the convex sum condition (14) yields to

$$\dot{x} = \sum_{i=1}^{N_r}\sum_{j=1}^{N_r} h_i(z)h_j(z)\big((A_i - B_i K_j)x + B_i F_j v\big)\,. \tag{23}$$

Inserting of (23) into the derivative of the Lyapunov function (6) results directly in

$$
\begin{aligned}
\dot{V}(x) = & \sum_{i=1}^{N_r}\sum_{j=1}^{N_r} h_i(z)h_j(z)\big((A_i - B_iK_j)x + B_iF_jv\big)^T Px \\
& + \sum_{i=1}^{N_r}\sum_{j=1}^{N_r} h_i(z)h_j(z)\, x^T P\big((A_i - B_iK_j)x + B_iF_jv\big) \; .
\end{aligned}
\tag{24}
$$

After few steps, we obtain

$$
\dot{V}(x) = \begin{pmatrix} x \\ v \end{pmatrix}^T \sum_{i=0}^{N_r}\sum_{j=1}^{N_r} h_i(z)h_j(z) \begin{pmatrix} PA_i + A_i^T P - PB_iK_j - K_j^T B_i^T P & PB_iF_j \\ F_j^T B_i^T P & 0 \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix} \; . \tag{25}
$$

As in the analysis of Section 3.2, the design criterion is derived by examining the QSR supply rate in (4):

$$
f_s(v,y) = \begin{pmatrix} y \\ v \end{pmatrix}^T \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \begin{pmatrix} y \\ v \end{pmatrix}
$$

Substitution of y by the output equation of (21) yields after few calculation steps

$$
f_s(v,y) = \begin{pmatrix} x \\ v \end{pmatrix}^T \sum_{i=1}^{N_r} h_i(z) \begin{pmatrix} C_i^T QC_i & C_i^T S \\ F_j^T B_i^T P - S^T C_i & -R \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix} \; . \tag{26}
$$

With (25) and (26), the result for the inequality (4) becomes

$$
\begin{aligned}
& \begin{pmatrix} x \\ v \end{pmatrix}^T \sum_{i=1}^{N_r}\sum_{i=1}^{N_r} h_i(z)h_j(z) \\
& \qquad \cdot \underbrace{\begin{pmatrix} PA_i + A_i^T P - PB_iK_j - K_j^T B_i^T P - C_i^T QC_i & PB_iF_j - C_i^T S \\ F_j^T B_i^T P - S^T C_i & -R \end{pmatrix}}_{\Gamma_{ij}(P,K_j,F_j) \preccurlyeq 0} \begin{pmatrix} x \\ v \end{pmatrix}^T \leq 0
\end{aligned}
\tag{27}
$$

Note that the matrix $\Gamma_{ij}(P,K_j,F_j)$ is not linear in P,K_j and F_j. With the restriction that only systems with directly measurable states $y = I_p x$ are to be analyzed regard to passive, strictly passive, and $\mathcal{L}_2$-gain systems we obtain $\Gamma_{ij}(X,X^2,K_j,F_j)$. The matrix is therefore linear in K_j and F_j, but bilinear in X. Steps required for this are explained below. Application of the congruence equivalent relation for (27) gives

$$\Gamma_{ij}(P,K_j,F_j) \preccurlyeq 0 \quad \Leftrightarrow \quad W\Gamma_{ij}(P,K_j,F_j)W^T \preccurlyeq 0 \tag{28}$$

with

$$W = \begin{pmatrix} X & 0 \\ 0 & X \end{pmatrix}, \quad X := P^{-1},, \quad X = X^T \tag{29}$$

yields

$$W\,\Gamma_{ij}(P,K_j,F_j)\,W^T = \\ \begin{pmatrix} A_iX + XA_i^T - B_iK_jX - XK_j^TB_i^T - XC_i^T\,QC_iX & B_iF_jX - X\,C_i^T\,SX \\ XF_j^TB_i^T - XS^T\,C_iX & -XRX \end{pmatrix}.$$

Utilizing the new variables $M_j := K_jX$, $N_j := F_jX$, where $M_j^T = X^TK_j^T = XK_j$ and $N_j^T = XF_j^T$, we obtain for $C_i = I_p$ the relaxed bilinear matrix inequalities

$$\begin{aligned} & X \succ 0\,, \quad \Gamma_{ij}^{\alpha}(X,M_j,N_j) + \Gamma_{ji}^{\alpha}(X,M_j,N_j) \preccurlyeq 0\,, \\ & \Gamma_{ii}^{\alpha}(X,M_j,N_j) \preccurlyeq 0 \quad \text{for all} \quad i = 1,2,\dots,N_r\,, \quad j = i+1, i+2,\dots,N_r \\ & \qquad\qquad s.t. \quad h_i(z)h_j(z) \neq 0,\ \exists z \end{aligned} \tag{30}$$

with $\alpha = \{a,b,c\}$. The passivity-based controller synthesis for the three types of passivity introduced in Section 2 is formalized by (30) for

- passive systems: $Q = 0_{p,p}$, $S = \frac{1}{2}I_p$, $R = 0_{p,p}$ with

$$\Gamma_{ij}^{a}(X,M_j,N_j) = \begin{pmatrix} A_iX + XA_i^T - B_iM_j - M_j^TB_i^T & B_iN_j - \frac{1}{2}X^2 \\ N_jB_i^T - \frac{1}{2}X^2 & 0 \end{pmatrix} \tag{31}$$

- strictly passive systems: $Q = -\varepsilon I_p, S = \frac{1}{2} I_p, R = -\delta I_p$ with

$$\Gamma^b_{ij}(X, M_j, N_j) = \begin{pmatrix} A_i X + X A_i^T - B_i M_j - M_j^T B_i^T + \varepsilon X^2 & B_i N_j - \frac{1}{2} X^2 \\ N_j^T B_i^T - \frac{1}{2} X^2 & \delta X^2 \end{pmatrix} \tag{32}$$

- $\mathcal{L}_2$-gain systems $Q = -I_p, S = 0_{p,m}, R = \gamma I_m$ with

$$\Gamma^c_{ij}(X, M_j, N_j) = \begin{pmatrix} A_i X + X A_i^T - B_i M_j - M_j^T B_i^T + X^2 & B_i N_j \\ N_j^T B_i^T & -\gamma X^2 \end{pmatrix} \tag{33}$$

Note the state feedback gain K_j and feed forward gain F_j for $j = 1, 2, .., N_r$ of the PDC control law (22) are calculated by $K_j = M_j X^{-1}$ and $F_j = N_j X^{-1}$.

5 Illustrating Example

Finally, after formal consideration of the QSR-dissipativity applied to T-S systems, an illustrative mechanical toy example is examined. This was selected in such a way that an analytical solution is also available. Given is a mechanical oscillator with one degree of freedom shown in Figure 1. The displacement of the rigid body with $m = 2$ (kg) is restricted to the direction x (m), whereby the stiffness in the mechanical system is described via the spring $c = 2$ (N/m) and the damping is described with the constituent formula

$$d(\dot{x}) = d_0 + d_1 \dot{x}^2 \tag{34}$$

with the parameters $d_0 = 0.1$ (N/m/s) and $d_1 = 0.05$ (N/m^2/s^2). The state space model

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{c}{m} x_1 - \frac{d_0}{m} x_2 - \frac{d_1}{m} x_2^3 + \frac{1}{m} u, \qquad x_0 = x(0), \\ y &= Cx \end{aligned} \tag{35}$$

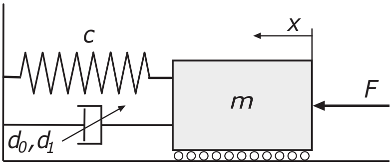


Figure 1: Mechanical Oscillator with nonlinear damper as an illustrating example

with $x = (x_1\,,x_2)^T := (x\,,\dot{x})^T$ and $u := F$ provides a complete description of the system dynamics. Let us first examine the passivity property of (35) analytically by using the dissipation inequality (1) directly i.e. without the proposed LMI formulation. The following Lyapunov function candidate is used for the example:

$$\begin{aligned} V(x) &= \frac{1}{2}cx_1^2 + \frac{1}{2}mx_2^2 > 0 \quad \text{for} \quad x_1,x_2 \neq 0 \\ \dot{V}(x) &= c\,x_1\,\dot{x_1} + m x_2\,\dot{x_2} \end{aligned}$$

Substituting the derivatives with the right hand sides of (35) results in

$$\dot{V}(x) = c\,x_1\,x_2 + m x_2\big(-\frac{c}{m}x_1 - \frac{d_0}{m}x_2 - \frac{d_1}{m}x_2^3 + \frac{1}{m}u\big) = -d_0\,x_2^2 - d_1\,x_2^4 + x_2 u\,.$$

Note, if $u = 0$ then $\dot{V}(x) < 0$ for all $x_2 \neq 0$ holds, this means that $V(x)$ is a Lyapunov function and the eigen-motion of (35) is asymptotically stable [5].

With an input $u \neq 0$ the derivative of dissipation inequality (1) results in

$$\dot{V}(x) \leq f_s(u,y) = u^T y\,. \tag{36}$$

Choosing the supply rate of $f_s(u,y) = u^T y$, the fulfillment of the inequality (36) directly results in the evidence of system passivity. Let us consider two orthogonal cases to investigate the passivity of the mechanical oscillator. In the first case only the position x and in the second case the velocity $\dot{x}$ can be measured, i.e. the output matrix C in (35) is

- Case 1: $C = (1\,,0) \quad \Rightarrow \quad y = x_1 = x\,, \quad f_s(u,y) = u\,x_1$

- Case 2: $C = (0\,,\,1) \quad \Rightarrow \quad y = x_2 = \dot{x} \quad f_s(u,y) = u\,x_2$

Thus, for the first case the dissipation inequality (36) for passive systems is

$$-d_0\,x_2^2 - d_1\,x_2^4 + x_2 u \;\leq\; u\,x_1 \;\Rightarrow\; \text{ infeasible problem for } x_1, x_2 \in \mathbb{R}$$

and for the second case

$$\begin{aligned} -d_0\,x_2^2 - d_1\,x_2^4 + x_2 u \;&\leq\; u\,x_2 \\ -d_0\,x_2^2 - d_1\,x_2^4 \;&\leq\; 0 \Rightarrow \text{ feasible problem for } x_1, x_2 \in \mathbb{R} \end{aligned}$$

The results are independent from the values of d_0 and d_1. Furthermore, the choice of output has a significant influence on the system characteristics. This means that the system properties of passivity not only depend on the input to state mapping but also on the effect of internal systems states on the output.
After the analytical investigation, the LMI-based criteria is now implemented by numerical convex optimization. First the nonlinear system is presented as T-S system. For this purpose, the nonlinear sector approaches is used, where T-S models exactly represent nonlinear systems in a bounded state space. In the case study, the nonlinearity is determined by the interval of $z \in [z_{min}, z_{max}]$ with $z := \dot{x}^2$, $z_{min} = 0$ and $z_{max} = \dot{x}^2_{max}$. The damping function $d(z) = d_0 + d_1 z$ proposed in (34) with the new coordinate z can therefore be substituted by the following sector function

$$d(z) = \frac{d_{max} - d(z)}{d_{max} - d_{min}}\,d_{min} + \frac{d(z) - d_{min}}{d_{max} - d_{min}}\,d_{max} = h_1(z)\,d_{min} + h_2(z)\,d_{max} \qquad (37)$$

with $d(z) \in [d_{min}, d_{max}]$, where $d_{min} = d_0$ and $d_{max} = d_0 + d_1\,z_{max}$. The membership functions of the T-S models result from the z-dependent weightings of the sector bounds in (37)

$$h_1(z) = \frac{d_{max} - d(z)}{d_{max} - d_{min}}\,, \qquad h_2(z) = \frac{d(z) - d_{min}}{d_{max} - d_{min}}\;. \qquad (38)$$

The state space model of the mechanical oscillator (35) then results as follows

$$
\begin{aligned}
\dot{x} &= \begin{pmatrix} 0 & 1 \\ -\frac{c}{m} & -\frac{d(z)}{m} \end{pmatrix} x + \begin{pmatrix} 0 \\ -\frac{1}{m} \end{pmatrix} u \\
&= \begin{pmatrix} 0 & 1 \\ -\frac{c}{m} & -h_1(z)\,\frac{d_{min}}{m} - h_2(z)\,\frac{d_{max}}{m} \end{pmatrix} x + \begin{pmatrix} 0 \\ -\frac{1}{m} \end{pmatrix} u \,.
\end{aligned}
$$

Utilizing the convex sum property (14), here with $h_1(z) + h_1(z) = 1 \;\; \forall z$, we get the final T-S representation of (35) to

$$
\dot{x} = h_1(z) \underbrace{\begin{pmatrix} 0 & 1 \\ -\frac{c}{m} & \frac{d_{min}}{m} \end{pmatrix}}_{A_1} x + h_2(z) \underbrace{\begin{pmatrix} 0 & 1 \\ -\frac{c}{m} & \frac{d_{max}}{m} \end{pmatrix}}_{A_2} + \underbrace{\begin{pmatrix} 0 \\ -\frac{1}{m} \end{pmatrix}}_{B} u \,. \tag{39}
$$

and in the standard sum form with the output equation

$$
\dot{x} = \sum_{i=1}^{2} h_i(z) A_i + Bu, \qquad y = Cx \,. \tag{40}
$$

Note that B, C, D are common and $D = 0$. That significantly reduce the number of LMIs to verify the QSR-dissipativity by (20) of T-S systems. With the specification $Q = 0_{p,p}$, $S = \frac{1}{2} I_p$, $R = 0_{p,p}$, which is used to examine only the passivity as in the analytical analysis, we obtain the reduced condition to be verified

$$
\begin{aligned}
& P \succ 0 \,, \\
& \Gamma_i(P) \preccurlyeq 0 \quad \text{for all} \quad i = 1, 2, \ldots, N_r \,,
\end{aligned} \tag{41}
$$

where

$$
\Gamma_i(P) = \begin{pmatrix} PA_i + A_i^T P & PB - \frac{1}{2} C^T \\ B^T P - \frac{1}{2} C & 0 \end{pmatrix} . \tag{42}
$$

Using the numerical values of the model parameters given above, the following results for (41), (42) were obtained with the SDPT3-4 solver within the YALMIP toolbox (version 20210331), Matlab R2020a:

- Case 1: $C=(1,0) \quad \Rightarrow$ `Infeasible problem (SDPT3-4)`
- Case 2: $C=(0,1) \quad \Rightarrow$ `Successfully solved (SDPT3-4)`

The result refers to the sector limit $z_{max} = \dot{x}^2_{max}$ with upper bound $\dot{x}_{max} = 2$. But could also be confirmed for larger values such as $\dot{x} = \{2,4,\dots,18,20\}$ i.e. the distance between the linear models in the parameter space is up two orders larger.

6 Conclusion

The QSR-dissipativity for the class of T-S fuzzy systems was investigated. An LMI-based verification and a BMI-based design approach for passive-based control were presented. Studies on synthesis with BMIs have yet to be carried out. A two-step procedure could convert the latter BMI into an LMI formulation. First, a Lyapunov function is calculated with the left hand side of the standard dissipation inequality, and in the second step, X^2 is calculated using an auxiliary variable. What could be shown in closing is that the proof of passivity in T-S fuzzy systems is numerically possible by means of convex optimization algorithms as implemented in [6]. Furthermore, because only simple test models have been used to date, more realistic physical process models [11], such as those used to verify the stability of distributed power systems, are also being investigated using the proposed approach.

References

[1] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, Philadelphia, PA, USA: SIAM, 1994.

[2] G. Feng, *A Survey on Analysis and Design of Model-Based Fuzzy Control Systems*, IEEE Transaction on Fuzzy Systems, vol. 14, no. 5, October 2006.

[3] A.T. Nguyen, T. Taniguchi, L. Eciolaza, M. Sugeno. *Fuzzy Control Systems: Past, Present and Future*, IEEE Computational Intelligence Magazine, vol: 14, no. 1, February 2019.

[4] H. Ohtake, K. Tanaka, and H. O. Wang. *Fuzzy Modeling via Sector Nonlinearity Concept*. In Joint 9th IFSA World Congress and 20th NAFIPS International Conference, pp. 127-132, Vancouver, Canada, 2001.

[5] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

[6] J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In Proceedings of the IEEE CACSD, pages 284–289, Taipei, Taiwan, 2004.

[7] H. J. Marquez, Nonlinear Control Systems, John Wiley & Sons Lt, 2003

[8] K. Tanaka and H.O. Wang. *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. John Wiley & Sons, Inc, 2001.

[9] A. van der Schaft, L2-Gain and Passivity Techniques in Nonlinear Control. Cham, Switzerland: Springer International Publishing, 2017.

[10] M. Arcak, C. Meissen, and A. Packard, Networks of Dissipative Systems - Compositional Certification of Stability, Performance, and Safety. AG Switzerland: Springer Nature, 2016.

[11] H. Schulte, Advanced Control of Grid-Integrated Renewable Energy Power Plants: LMI-based Design in the Takagi-Sugeno Framework, Wiley-IEEE Press, John Wiley & Sons 2024, Hoboken, New Jersey, 2024

Robust Reconstruction of Grid Frequency in Disturbed Converter-Based Power Systems by Takagi-Sugeno Fuzzy Approach

Nico Goldschmidt[1], Horst Schulte[1], Ignacio Glenny Crende[2]

[1] University of Applied Sciences (HTW) Berlin,
Faculty 1: School of Engineering - Energy and Information,
Wilhelminenhofstr. 75A, 12459 Berlin, Germany,
Email: schulte@htw-berlin.de

[2] Universitat Politècnica de Catalunya
Department d'Enginyeria Elèctrica (CITCEA-UPC)
UPC. Av. Diagonal 647, Pl. 2, Barcelona, 08028, Spain
Email: ignacio.glenny@upc.edu

Summary

This study investigates the conversion of the proposed phase-locked loop design into the Takagi-Sugeno fuzzy model approach. Here, the Takagi-Sugeno fuzzy model is used to describe the nonlinear dynamics of the PLL system through fuzzy rules that summarize locally linear submodels. By applying the fuzzy model, the control performance of the PLL system can be further improved, and adaptability to different operating conditions can be enhanced. The work shows how state feedback and pole placement can be integrated into the Takagi-Sugeno fuzzy framework, enabling robust and precise frequency estimation under disturbed conditions. Simulations illustrate the effectiveness of the combined approach and highlight its potential for application in power systems.

1 Introduction

The continuous expansion of renewable energy in power grids aims for 100% renewable energy production, necessitating their integration as active control units for grid stabilization. Reducing conventional energy producers introduces more disturbances and harmonics, resulting in higher frequency and voltage variations. Precise detection of fundamental symmetrical components is crucial for the reliable operation of inverter-based renewable energy sources, even in disrupted areas.

This work focuses on exact frequency reconstruction in converter-based three-phase grids. A key challenge is the reliable reconstruction of heavily loaded systems due to harmonics and other disturbances. Unlike traditional phase-locked loops that filter out disturbances, the method by [1] employs a recursive estimation process. First, it reconstructs the fundamental oscillation from the disturbed signal, then it reconstructs the grid frequency using a model-based phase-locked loop (PLL). This method has proven effective even in highly disturbed grids. Previous work by [2] introduced a linear model-based I-extended state feedback, achieving fast and precise results, particularly in 50Hz grids, and effective performance with larger frequency deviations. The community's curiosity about future systems operating at non-fixed frequencies led to exploring an extension using nonlinear control methods.

The extension in this paper relies on the recursive estimation method for fundamental oscillation reconstruction, serving as a base signal for frequency reconstruction. For final frequency reconstruction and deviating operating frequencies, a Takagi-Sugeno Fuzzy model [3] is used. This model simplifies the nonlinear dynamics of the PLL into local linear submodels, achieving the desired operating range. The previous model-based state controllers and pole assignment methods are combined with the Takagi-Sugeno approach using Linear Matrix Inequalities (LMI).

The work is structured as follows: Section 2 reviews the modeling from previous work, setting the stage for the extension in Section 3. Stability is ensured using the Takagi-Sugeno method. Section 4 presents simulation results, followed by the conclusion in Section 5.

2 Modeling

In the paper [2] published in 2020, the systematic design of an I-extended state controller for a phase-locked loop was presented, which reconstructs the grid angle based on an estimated fundamental oscillation of a disturbed grid using a Least Mean Square estimator based on [1]. The formal basis of PLL modeling is provided by the synchronous voltage reference frame in *d*-*q* coordinates. The corresponding coordinators $\mathbf{v}_{dq} = (v_d \quad v_q)^T$ are calculated from the symmetric components of the three phase system $\mathbf{v}_{abc} = (v_a \, v_b \, v_c)^T$ by the Clark and Park transformation

$$\begin{pmatrix} v_d \\ v_q \end{pmatrix} = \underbrace{\frac{2}{3} \begin{pmatrix} \cos(\widehat{\varphi}) & \cos(\widehat{\varphi} - \frac{2}{3}\pi) & \cos(\widehat{\varphi} + \frac{2}{3}\pi) \\ -\sin(\widehat{\varphi}) & -\sin(\widehat{\varphi} - \frac{2}{3}\pi) & -\sin(\widehat{\varphi} + \frac{2}{3}\pi) \end{pmatrix}}_{\mathbf{T}_{dq}(\widehat{\varphi})} \begin{pmatrix} v_a \\ v_b \\ v_c \end{pmatrix} , \qquad (1)$$

where $\widehat{\varphi}$ denotes the estimated phase angle. It should be noted that a synchronous coordinate system can be formed with the Clark and Park transformation (1) if the three-phase system is symmetric. Thus (1) can be applied to a unbalanced system by transforming $\mathbf{v}_{abc} = (v_a \, v_b \, v_c)^T$ in symmetric components with

$$\mathbf{v}_{abc} = \mathbf{v}_{abc}^{+} + \mathbf{v}_{abc}^{-} + \mathbf{v}_{abc}^{0} \qquad (2)$$

with the positive sequence components $\mathbf{v}_x^+$, negative sequence components $\mathbf{v}_x^-$, and zero sequence components $\mathbf{v}_x^0$. Note that the zero components are not taken into account because the park transformation yields a zero vector. By substituting follows with (1) and the addition theorem

$$\begin{aligned} v_d &= \underbrace{-V^+ \sin(\Delta\widehat{\varphi}^+)}_{v_d^+} + \underbrace{V^- \sin(\Delta\widehat{\varphi}^+ - \varphi_n)}_{v_d^-} \\ v_q &= \underbrace{V^+ \cos(\Delta\widehat{\varphi}^+)}_{v_q^+} - \underbrace{V^- \cos(\Delta\widehat{\varphi}^+ - \varphi_n)}_{v_q^-} , \end{aligned} \qquad (3)$$

where $\Delta\widehat{\varphi}^{+}$ denotes the difference between of the reconstructed and true phase angle

$$\Delta\widehat{\varphi}^{+} = \varphi^{+} - \widehat{\varphi} \quad . \tag{4}$$

The derivation of (3) leads to

$$\begin{array}{rcrl} \dot{v}_d^{+} & = & - & \Delta\widehat{\omega}^{+} V^{+} \cos(\Delta\widehat{\varphi}^{+}) \\ \dot{v}_q^{+} & = & - & \Delta\widehat{\omega}^{+} V^{+} \sin(\Delta\widehat{\varphi}^{+}) \\ \dot{v}_d^{-} & = & & (\Delta\widehat{\omega}^{+} - \omega_n)\, V^{-} \cos(\Delta\widehat{\varphi}^{+} - \varphi_n) \\ \dot{v}_q^{-} & = & & (\Delta\widehat{\omega}^{+} - \omega_n)\, V^{-} \sin(\Delta\widehat{\varphi}^{+} - \varphi_n) \end{array} \tag{5}$$

with

$$\Delta\widehat{\omega}^{+} = \Delta\dot{\widehat{\varphi}}^{+} \quad , \qquad \omega_n = \dot{\varphi}_n >> 0 \quad . \tag{6}$$

The differential equation system is simplified by inserting (3) into (5). In summary, this results in

$$\begin{pmatrix} \dot{v}_d^{+} \\ \dot{v}_q^{+} \\ \dot{v}_d^{-} \\ \dot{v}_q^{-} \end{pmatrix} = \begin{pmatrix} -\Delta\widehat{\omega}^{+} v_q^{+} \\ \Delta\widehat{\omega}^{+} v_d^{+} \\ -\Delta\widehat{\omega}^{+} v_q^{-} \\ \Delta\widehat{\omega}^{+} v_d^{-} \end{pmatrix} \quad . \tag{7}$$

For the model-oriented control design it is necessary to transform (7) in the state-space form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) = \begin{pmatrix} 0 & -u & 0 & 0 \\ u & 0 & 0 & 0 \\ 0 & 0 & 0 & -u \\ 0 & 0 & u & 0 \end{pmatrix} \mathbf{x} = \mathbf{A}(u)\mathbf{x} \tag{8}$$

with the state vector

$$\mathbf{x} = \left(\mathbf{x}^{+}, , \mathbf{x}^{-}\right)^{T} = \left(v_d^{+}, v_q^{+}, v_d^{-}, v_q^{-}\right)^{T} , \tag{9}$$

the output vector $\mathbf{y}$ and scalar valued input u, i.e.

$$\mathbf{y} = \mathbf{x}, \quad u = \Delta\widehat{\omega}^{+} \ . \tag{10}$$

It is interesting to note that the deduced nonlinear state-space model (8) has no affine input u but appears as a variable parameter in the system matrix $\mathbf{A}(u)$.

2.1 State Space Feedback Design for Setpoint Sequences with Pole Specification

Based on the Taylor-linearized model in (8) a systematic LMI-based design according to [4], [5], [6] is presented for further use in the entire PLL synthesis. In contrast to classical PI controller design the approach enables a pole placement in a desired region in the complex plane by LMI-constraints. For stationary exactness the system will be extended to a setpoint control for $\mathbf{y}_{ref} \neq \mathbf{0}$. In order to be able to guarantee the stability of the closed loop, the integrator eigenvalues must be controllable. According to the criterion of Hautus, the r integral eigenvalues $\lambda = 0$ can be controlled, if and only if the condition

$$\text{Rank} \begin{pmatrix} \mathbf{A}_c & \mathbf{0} & | & \mathbf{b}_c \\ \mathbf{c}_c^T & 0 & | & d_c \end{pmatrix} = n + r \tag{11}$$

is fulfilled. For the estimation of the actual angular frequency value with the synchronous reference frame approach the exact regulation of the positive q component to ZERO is necessary, so the output matrix $\mathbf{c}_c$ is defined by

$$\mathbf{c}_c^T = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix} \ . \tag{12}$$

Taking into account the output matrix $\mathbf{c}_c$, we get the extended system and input matrix

$$\tilde{\mathbf{A}}_c = \begin{pmatrix} \mathbf{A}_c & \mathbf{0} \\ -\mathbf{c}_c^T & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \qquad \tilde{\mathbf{b}}_c = \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix} \in \mathbb{R}^{(n+1)\times m} \ . \tag{13}$$

The setpoint control law can then be specified for the extended system with the input

$$\Delta\tilde{u} \quad = \quad -\begin{pmatrix}\mathbf{k}_x & -k_I\end{pmatrix} \tag{14}$$

and

$$\Delta r = \int_0^t \Delta e \, d\tau \,, \qquad \Delta e = \Delta y_{ref} - \mathbf{c}_c^T \Delta\mathbf{x} \tag{15}$$

as follows

$$\Delta\tilde{u} \quad = \quad -\mathbf{k}_x^T \Delta\mathbf{x} + k_I \int \Delta \quad . \tag{16}$$

For the sake of clarity, previous results are summarized in Fig. 1. The block

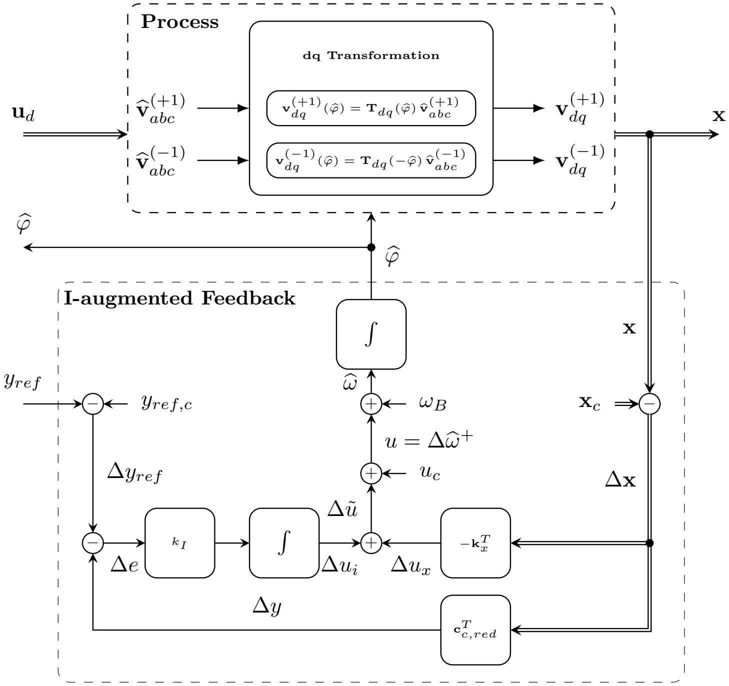


Figure 1: Block diagram of the I-augmented feedback control

diagram illustrates the I-augmented state feedback structure including the transformation of a three phase system into a synchronous reference frame for positive and negative sequences. Based on the extended uncertain system of (8)

$$\Delta\dot{\tilde{\mathbf{x}}} \quad = \quad \tilde{\mathbf{A}}_c \Delta\tilde{\mathbf{x}} + \tilde{\mathbf{b}}_c \Delta\tilde{u} \tag{17}$$

with the equilibrium point

$$\mathbf{x}_c \quad = \quad [1, 0, 0, 0]^T \ , \ u_c \quad = \quad 0 \ , \tag{18}$$

$\mathbf{A}_c$ as the system matrix and $\mathbf{b}_c$ as the input vector, the objective is to find a state variable feedback

$$\Delta\tilde{u} \quad = \quad -\tilde{\mathbf{k}}^T \Delta\tilde{\mathbf{x}} \tag{19}$$

such that the closed-loop eigenvalues of $\left(\tilde{\mathbf{A}} - \tilde{\mathbf{b}}\tilde{\mathbf{k}}^T\right)$ are in $S(\alpha_{min}, \alpha_{max})$. The so-defined pole region $S(\alpha_{min}, \alpha_{max})$ is shown in Fig. 2. The constraint pre-

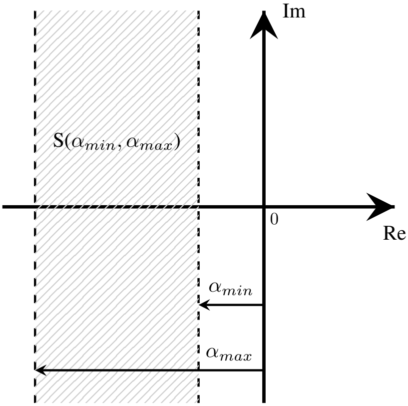


Figure 2: Pole region S (α_{min}, α_{max})

defined region of closed loop eigenvalues $S(\alpha_{min}, \alpha_{max})$ guarantees a desired performance specified by an maximum overshoot, the frequency range of the damped oscillations, rise time, and settling time as shown in [4]. However, it is assumed that there is a feasible solution to the LMI condition (22). The system

(17) controlled by state feedback (19) is exponentially stable in respective to the defined region $S(\alpha_{min}, \alpha_{max})$ if there exists a Lyapunov function

$$\mathbf{V}(\Delta\tilde{\mathbf{x}}) \quad = \quad \Delta\tilde{\mathbf{x}}^T\mathbf{P}\Delta\tilde{\mathbf{x}},\ \mathbf{P} > 0 \tag{20}$$

with $\mathbf{P} = \mathbf{P}^T \in \mathbb{R}^{n+r\times n+r}$ and $\mathbf{X} = \mathbf{P}^{-1}$ that fulfills with respect to

$$\mathbf{P}\mathbf{A}_c^T + \mathbf{A}_c\mathbf{P} < \mathbf{b}_c\mathbf{m} + \mathbf{m}^T\mathbf{b}_c^T \qquad , \tag{21}$$

where and $\mathbf{m} \in \mathbb{R}^{1\times n}$. The LMI stability are given by

$$\begin{aligned}
-2\alpha_{max}\mathbf{X} \quad &< \quad \tilde{\mathbf{A}}_c\mathbf{X} + \mathbf{X}\tilde{\mathbf{A}}_c^T - \tilde{\mathbf{b}}_c\mathbf{m} - \mathbf{m}^T\tilde{\mathbf{b}}_c^T \quad < \quad -2\alpha_{min}\mathbf{X} \\
\begin{pmatrix} -r\mathbf{P} & \mathbf{A}\mathbf{P} - \mathbf{b}\mathbf{m} \\ \mathbf{P}\mathbf{A}^T - \mathbf{m}^T\mathbf{b}^T & -r\mathbf{P} \end{pmatrix} \quad &< \quad 0 \\
(\mathbf{E}\otimes\mathbf{A})\,\mathbf{P} + \mathbf{P}\,(\mathbf{E}\otimes\mathbf{A})^T - (\mathbf{E}\otimes\mathbf{b})\,\mathbf{m} - \mathbf{m}^T\,(\mathbf{E}\otimes\mathbf{b})^T \quad &< \quad 0
\end{aligned} \tag{22}$$

with

$$\mathbf{E} = \begin{pmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{pmatrix} \qquad , \tag{23}$$

with $\alpha_{max} > \alpha_{min} > 0$, where $\mathbf{m} \in \mathbb{R}^{1\times n+r}$. As described in [4] and [5]

$$\begin{pmatrix} \sin\theta\,(\mathbf{AX} + \mathbf{XA}^T - \mathbf{M}^T\mathbf{B}^T - \mathbf{BM}) & \cos\theta\,(\mathbf{AX} - \mathbf{XA}^T + \mathbf{M}^T\mathbf{B}^T - \mathbf{BM}) \\ \cos\theta\,(-\mathbf{AX} + \mathbf{XA}^T - \mathbf{M}^T\mathbf{B}^T + \mathbf{BM}) & \sin\theta\,(\mathbf{AX} + \mathbf{XA}^T - \mathbf{M}^T\mathbf{B}^T - \mathbf{BM}) \end{pmatrix} < 0 \tag{24}$$

this results in the state feedback matrix

$$\tilde{\mathbf{k}}^T \quad = \quad \mathbf{m}\mathbf{X}^{-1} \tag{25}$$

for (19).

3 Takagi-Sugeno Control with Prefilter Design

In this work, the fuzzy approximation of the nonlinear state-space model is formulated using the Takagi-Sugeno (TS) model structure. Originally introduced in the context of fuzzy systems by [3], TS models are weighted combinations of linear or affine sub-models. These models can be derived either from input-output data using system identification techniques, as presented by [3], [7] and [8]. The general Takagi-Sugeno (TS) structure of a state-space model with so-called affine remainder terms is of the form:

$$\begin{aligned}\dot{\mathbf{x}} &= \sum_{i=1}^{N_r} h_i(\mathbf{z})(\mathbf{A}_i\mathbf{x}+\mathbf{B}_i\mathbf{u}+\mathbf{a}_i)\\ \mathbf{y} &= \sum_{i=1}^{N_r} h_i(\mathbf{z})(\mathbf{C}_i\mathbf{x}+\mathbf{D}_i\mathbf{u}+\mathbf{c}_i)\end{aligned} \tag{26}$$

where $\mathbf{z} \in \mathbb{R}^{N_l}$ denotes the vector of premise variables, $\mathbf{x} \in \mathbb{R}^n$ is the vector of system states, $\mathbf{u} \in \mathbb{R}^m$ is the input vector, $\mathbf{y} \in \mathbb{R}^p$ is the vector of outputs. The matrices $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i$, and $\mathbf{D}_i$, and the affine vectors $\mathbf{a}_i$ and $\mathbf{c}_i$, are of appropriate dimensions. The membership functions h_i fulfill the following conditions:

$$\begin{aligned}\sum_{i=1}^{N_r} \quad h_i(\mathbf{z}) &= 1, \quad \text{for} \quad i = 1,\dots,N_r,\\ h_i(\mathbf{z}) &\geq 0, \quad i = 1,\dots,N_r,\end{aligned} \tag{27}$$

where the above is known as the convex sum condition. The Takagi-Sugeno fuzzy approximation of the nonlinear state-space model (8) is now determined. First, the variables that govern the nonlinearities must be identified, as the scheduling vector should consist of these variables. This involves deciding on $\mathbf{z}$ as a selection of inputs, states, outputs, and external measurements. In this case, the scheduling vector is chosen based on the input

$$z = u = \Delta\widehat{\omega}^+ \ . \tag{28}$$

With frequency deviation $\Delta\hat{\omega}^+$ the frequencies f_{min}, f_{max}, and f_{ref} are defined as follows:

$$f_{min} \leq f_{ref} \leq f_{max} \quad \text{where} \quad 45Hz \leq 50Hz \leq 55Hz \quad . \tag{29}$$

The vector of equilibrium points $u_{c,i}$ for $i = 1,\dots,N_r$ is defined as follows:

$$\mathbf{u}_c = 2\pi\,(-5.0;-4.8;-4.6;\dots;u_{c,i};\dots;4.6;4.8;5.0)^T \ , \tag{30}$$

where $N_r = 51$ has been chosen as a good compromise between accuracy and performance. A Takagi-Sugneo fuzzy controller with prefilter in the following compact form is proposed

$$\Delta u \quad = \quad \mathbf{h}(z)\,\mathbf{V}\,\Delta\mathbf{x}_{\text{ref}} - \mathbf{h}(z)\,\mathbf{K}\,\Delta\mathbf{x}\,, \tag{31}$$

where the prefilter matrix is given

$$\mathbf{V} = \begin{pmatrix} v_{1,1} & v_{2,1} & v_{3,1} & v_{4,1} \\ \vdots & \vdots & \vdots & \vdots \\ v_{1,i} & v_{2,i} & v_{3,i} & v_{4,i} \\ \vdots & \vdots & \vdots & \vdots \\ v_{1,N_r} & v_{2,N_r} & v_{3,N_r} & v_{4,N_r} \end{pmatrix}$$

and the feedback gains are arranged as

$$\mathbf{K} = \begin{pmatrix} k_{1,1} & k_{2,1} & k_{3,1} & k_{4,1} \\ \vdots & \vdots & \vdots & \vdots \\ k_{1,i} & k_{2,i} & k_{3,i} & k_{4,i} \\ \vdots & \vdots & \vdots & \vdots \\ k_{1,N_r} & k_{2,N_r} & k_{3,N_r} & k_{4,N_r} \end{pmatrix} .$$

In combination with the row vector of all membership functions

$$\mathbf{h}(z) = (h_1(z)\; h_2(z)\; ....\; h_{N_r-1}(z)\; h_{N_r}(z)) \tag{32}$$

the formula (31) represents the compact matrix form of a Takagi-Sugeno control law with prefilter. To calculate the absolute output of (31), the values of the equilibrium points must still be taken into account by

$$u = \mathbf{h}(z)\,\mathbf{u}_c + \Delta u\,. \tag{33}$$

Obtaining the estimated phase for the positive sequence θ^+ after $\Delta\hat{\omega}^+ = u$ has been determined requires the equilibrium frequencies. The reconstructed frequency $\hat{f}$ is as follows:

$$\hat{f} \quad = \quad \mathbf{h}(z)\,\mathbf{f}_c^T - \Delta\hat{\omega}^+ \tag{34}$$

This reconstructed frequency $\hat{f}$ is then passed through an integrator in order to obtain the estimated phase θ^+.

4 Results and Discussion

In this paper, three primary events are used to compare the performance of the proposed TS prefilter PLL with an I-augemented PLL and a linear (static gain) prefilter PLL. The first simulated event is a phase-to-phase short circuit lasting 0.5 seconds. This event is evaluated in three different scenarios:

- Clean (Clean): No harmonics or measurement disturbances are present
- Low Harmonics (LH): Includes measurement disturbances
- High Harmonics (HH): Includes significant measurement disturbances

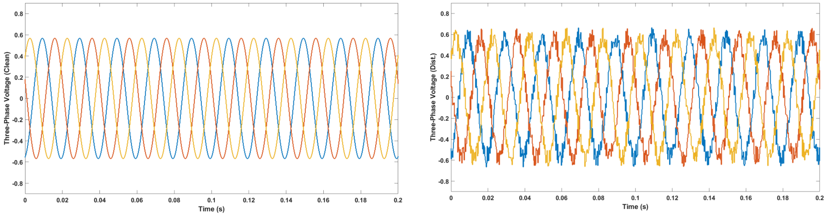
While the short circuit last for 0.5 seconds, the full simulation runs for 1 second. The second simulated event is a frequency variation (FreqVar) lasting 30 seconds. As in the short circuit, the three scenarios (clean, LH, and HH) are also considered here. The Table 1 shoes the key results of these simulations, including additional details about the various test conditions and scenarios.

Event	Short circuit			Freq. Variation		
	Clean	LH	HH	Clean	LH	HH
Duration (s)	1	1	1	30	30	30
Disturbance	Yes	Yes	Yes	Yes	Yes	Yes
5th Harmonic [%]	0.00	3.0	30.0	0.00	3.0	30.0
7th Harmonic [%]	0.0	1.5	15.0	0.0	1.5	15.0
V_d^+ SS Error [p.u.]	$7.4 \cdot 10^{-4}$	$7.4 \cdot 10^{-4}$	$7.4 \cdot 10^{-4}$	$4 \cdot 10^{-2}$	$4 \cdot 10^{-2}$	$4 \cdot 10^{-2}$
Settling Time [s]	0.038	0.038	0.038	-	-	-

Table 1: Key indicators with details about every simulation run

4.1 Disturbances and Harmonics

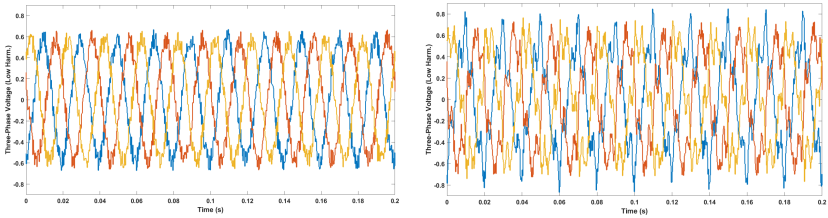
The disturbance is introduced as random noise in the measurements. Figure 3 (a) displays the three-phase voltage for the clean scenario, where no harmonics or measurement disturbances are present. In contrast, the effects of disturbance on the source voltage are illustrated in Figure 3 (b).



(a) (b)

Figure 3: 3ph voltage at the source without (a) and with (b) measurement disturbance

Figures 4 (a) and (b) show the disturbed 3ph system, which is intended to affect the PLL's ability to track the system's phase. No separate scenarios are considered solely for the disturbance. It is shown that the impact of the disturbance is negligible and it is included in the Low Harmonics and High Harmonics scenarios.

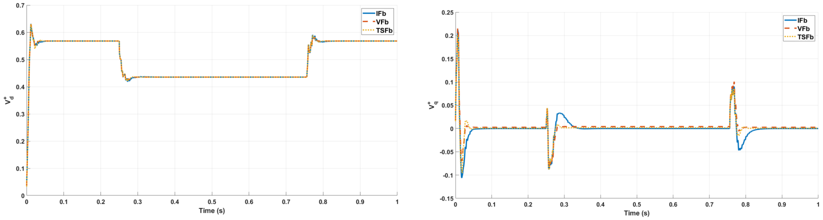


(a) (b)

Figure 4: Three phase voltage at the source for the Low Harmonics (a) and for the High Harmonics (b) scenario

4.2 Short Circuit scenario

Figure 5 (a), 5 (b) and Figure 7 illustrate the evolution of V_d^+, V_q^+, and Δf, respectively, throughout the entire short circuit simulation run under the High Harmonics scenario. In Figure 5, a voltage dip in V_d^+, V_q^+ can be observed, which measures some p.u. (a certain percentage of the steady-state value) as a result of phase-to-phase short circuit.

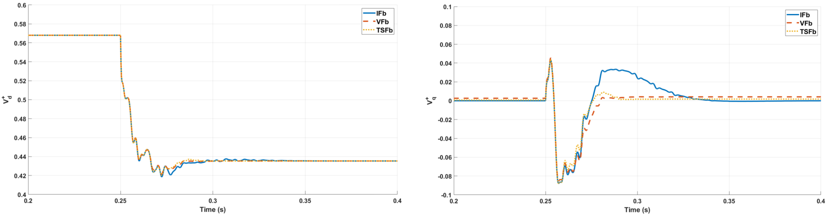


(a)

(b)

Figure 5: V_d^+ voltage (a) and V_q^+ voltage (b) for the short circuit High Harmonics scenario

The I-augmented PLL (IFB) requires more time to reach $V_q^+ = 0$ after the short circuit event; however, it is the only method that achieves this without any steady-state error. In contrast, the Static Prefilter (VFb) and the Takagi-Sugeno Static Prefilter (TSFb) exhibit steady-state errors of value and value p.u., respectively. This can be observed in Figure 5 (b) below.



(a)

(b)

Figure 6: Detail of V_d^+ voltage (a) and V_q^+ voltage (b) for the short circuit High Harmonics scenario

To illustrate the transition that systems undergo during the short-circuit event, the Figures 6 (a) and 6 (b) detail the transition period. The longer settling time of the I-augmented PLL (IFB) is more clearly visible in Figure 6 (a).

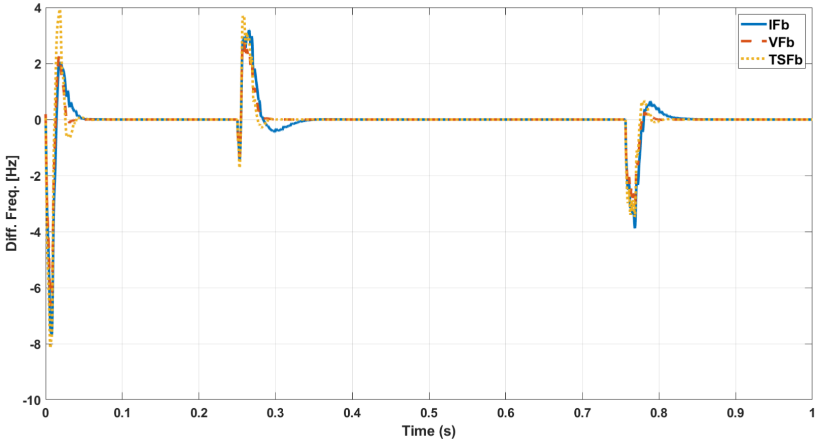
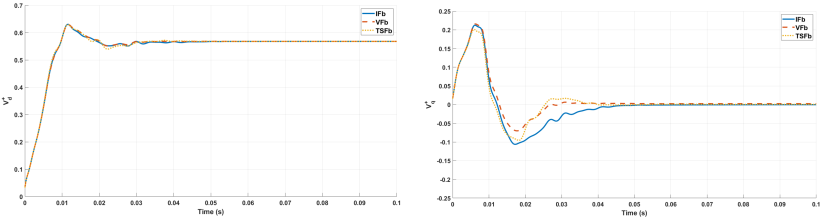


Figure 7: Δf for the short circuit High Harmonics scenario

The overshoot and longer settling times of the I-augmented PLL (IFB) in V_q^+ can be observed in Figure 6 (b), alongside the steady-state errors of the Static Prefilter (VFb) and the Takagi-Sugeno Prefilter (TSFb). All three PLL systems successfully reach the nominal steady-state frequency of 50 Hz, as shown in Figure 7. The settling time required by each of the PLLs to reach the target steady-state value, as seen in Table 1, is graphically represented in Figures 8 (a) and (b) below.



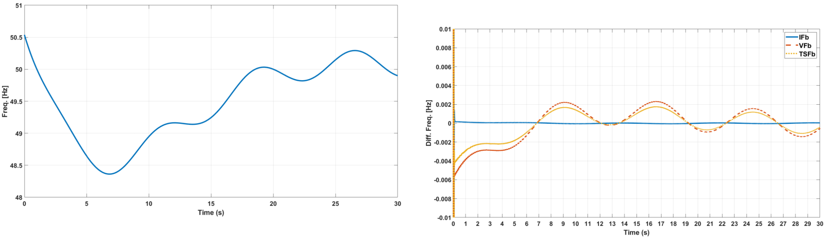
(a)

(b)

Figure 8: V_d^+ voltage (a) and V_q^+ voltage (b) in the initial transient for the short circuit High Harmonics scenario

4.3 Frequency Variation scenario

The evolution of the source frequency during the entire simulation run of frequency variation is shown in Figure 9 (a). All three PLL systems are able to reach different target steady-state frequencies. However, as seen in Figure 9 (b), the Static Prefilter (VFb) and Takagi-Sugeno (TSFb) PLL exhibit small steady-state errors ranging from 2 to -6 mHz.

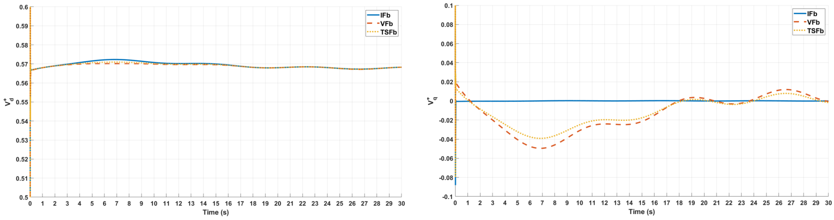


(a) Evolution of the original source frequenca during the Frequency Variation scenarion

(b) Δf for the short circuit High Harmonics scenario

Figure 9: Frequency variation scenarion

Figure 10 (a) and (b) illustrate the evolution of V_d^+ and V_q^+, respectively, throughout the simulation run of frequency variation for the high harmony scenario. A difference in the V_d^+ voltages among the three PLLs can be observed in Figure 10 (a).



(a) (b)

Figure 10: V_d^+ voltage (a) and V_q^+ voltage (b) for the short circuit High Harmonics scenario

The Static Prefilter (VFb) and the Takagi-Sugeno (TSFb) PLLs are unable to achieve $V_q^+ = 0$ when the frequency deviates from 50 Hz. This steady-state error is displayed in Figure 10 (b).

5 Conclusion

Using an electrical modeling and the I-augmented and Standard Prefilter PLL environments developed by [2], simulations of short-circuits, load jumps, and frequency variation events under various disturbance and harmonic scenarios were conducted. Short-circuit and frequency variation simulation results are included in this paper to demonstrate the phase-tracking capabilities of the TS Static Prefilter. The Least Mean Squares filter used to reconstruct the main harmonic is crucial for the results obtained by the various PLLs studied in this work, but its development is beyond the scope of this paper.

References

[1] X. Zhang, X. Cao, W. Wang, and C. Yun, Fault Ride-Through Study of Wind Turbines, Journal of Power and Energy Engineering, vol. 01, no. 05, pp. 25–29, 2013. [Online]. Available: `http://www.scirp.org/journal/doi.aspx?DOI=10.4236/jpee.2013.15004`

[2] N. Goldschmidt and H. Schulte, Estimation of Grid Frequency in Disturbed Converter-Based Power Systems by PLL State Variable Feedback, In: Proceedings of IFAC World Congress 2020, Berlin, Germany, 2020.

[3] T. Takagi and M. Sugeno, “Fuzzy Identification of Systems and Its Application to Modeling and Control,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[4] M. Chilali and P. Gahinet, “Hınfty design with pole placement constraints: An LMI approach,” *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 358–367, 1996.

[5] D. Arzelier, J. Bernussou, and G. Garcia, “Pole assignment of linear uncertain systems in a sector via a Lyapunov-type approach,” *IEEE Transactions on Automatic Control*, vol. 38, no. 7, pp. 1128–1132, 1993.

[6] S. Gutman and E. Jury, A general theory for matrix root-clustering in subregions of the complex plane, IEEE Transactions on Automatic Control, vol. 26, no. 4, pp. 853–863, Aug. 1981. [Online]. Available: `http://ieeexplore.ieee.org/document/1102764/`

[7] M. Sugeno and G. T. Kang, “Structure Identification of Fuzzy Models,” *Fuzzy Sets and Systems*, vol. 28, pp. 15–33, 1988.

[8] Z. Lendek, T. M. Guerra, R. Babuška, and B. de Schutter, *Stability Analysis and Nonlinear Observer Design Using Takagi-Sugeno Fuzzy Models*. Springer-Verlag Berlin Heidelberg, 2010.

[9] K. Tanaka, T. Ikeda, and H. O. Wang, “Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: quadratic stabilizability, H^\backslashinfty control theory, and linear matrix inequalities,” *IEEE Transactions on Fuzzy Systems*, vol. 4, pp. 1–13, 1996.

Investigating Reproducibility of Ultra-High Performance Concrete with Consistent Mechanical Properties: A Modeling Pipeline for Complex Manufacturing

Farzad Rezazadeh P.[1], Amin Abrishambaf[2], Axel Dürrbaum[1],
Gregor Zimmermann[2], Andreas Kroll[1]

[1]Department of Measurement and Control, University of Kassel
Mönchebergstr. 7, 34125 Kassel, Germany
E-Mail: farzad.rezazadeh, axel.duerrbaum, akroll@mrt.uni-kassel.de

[2]German Technologies & Engineering Concepts - G.tecz Engineering GmbH
Eichwaldstr. 38, 34125 Kassel, Germany
E-Mail: abrishambaf, zimmermann@gtecz.com

1 Introduction

Ultra-High Performance Concrete (UHPC) is distinguished by its exceptional mechanical properties and durability, offering significant advantages over conventional concrete [1]. However, achieving consistent UHPC quality remains a challenge, even when following the same formulation [2]. This inconsistency is largely due to the production process's sensitivity to variations in raw material properties, environmental conditions, and operational factors (see Figure 1). Adopting a holistic approach to the entire UHPC manufacturing process introduces high dimensionality and increases the cost of data generation, resulting in sparse datasets. To address these challenges, this study presents an automated modeling pipeline (see Figure 2) specifically designed to tackle high-dimensional and sparse data issues. The proposed pipeline comprises five layers: data generation based on a three-phase Design of Experiments (DoE), data preprocessing, ensemble-based feature importance determination [2], feature

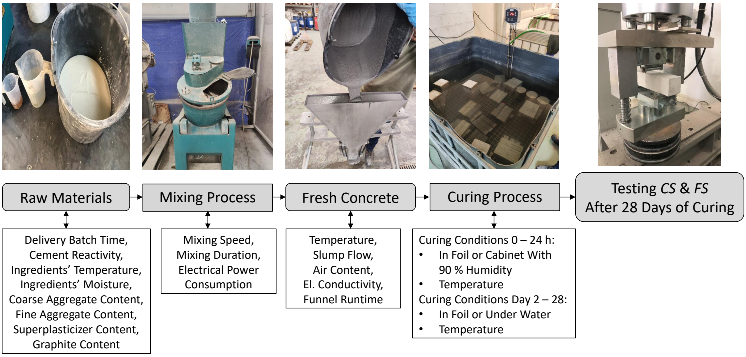


Figure 1: Ultra-High Performance Concrete (UHPC) Production Process [2]. The diagram shows key factors influencing UHPC production and testing. (El: Electrical, CS: Compressive Strength, FS: Flexural Strength)

selection using the Informed Non-Dominated Sorting Genetic Algorithm II (I-NSGA-II), and a modeling layer. The primary objective is to analyze the consistency of key mechanical properties, specifically compressive and flexural strength after 28 days of curing.

2 Modeling Pipeline for the Ultra-High Performance Concrete Production Process

2.1 Data Generation

Due to resource limitations restricting the number of experiments to 100 in this study, a feature pool was developed through a literature review and expert input, ensuring a comprehensive representation of the entire UHPC manufacturing process. To further manage the dimensionality of the input space, feature engineering was employed as an initial strategy to reduce the number of features while preserving critical information.

An effective DoE is crucial for managing complex input spaces. To address this, a three-phase DoE was developed: Screening, Optimal, and Complementary Phases. The L-50 Taguchi Orthogonal Array [3] was used in the Screening

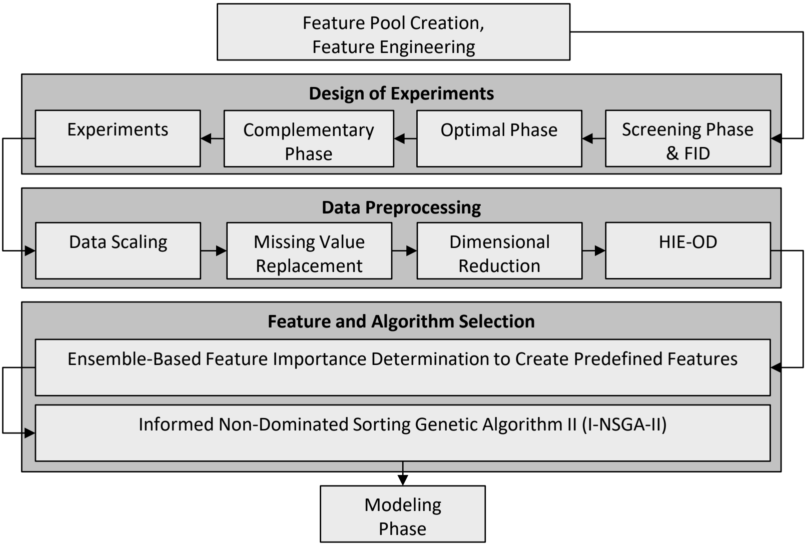


Figure 2: Automated Modeling Pipeline for Ultra-High Performance Concrete Manufacturing: An End-to-End Framework Covering Data Generation to Model Building. (FID: Feature Importance Determination, HIE-OD: Human-in-the-Loop Informed Ensemble-based Outlier Detection)

Phase due to its ability to handle high-dimensional spaces with few observations and ensure uniform coverage of the input space, resulting in 50 data points. This phase led to the removal of three non-essential features and the identification of curing conditions as the most critical feature.

In the Optimal Phase, the goal was to strategically position new data points relative to those established in the Screening Phase. To achieve this, 50 additional experiments were conducted using S-optimality criteria and Latin Hypercube Sampling (LHS) [4], refining the experimental design based on insights gained from the initial phase. Recognizing that curing conditions were the most influential factors, the Complementary Phase was conducted in parallel with the Optimal Phase. In this phase, each experiment designed in the Optimal Phase was replicated under two curing conditions: one based on the Optimal Phase design and the other informed by expert knowledge. This dual-condition approach generated two outputs per experiment, resulting in a total of 150 data points from 100 experiments. This comprehensive dataset ensures robust coverage of the UHPC manufacturing process for subsequent modeling.

2.2 Data Preprocessing

Given the high dimensionality and limited data size, it was crucial to reduce dimensionality while addressing missing values to maintain data integrity. Traditional outlier detection methods were unsuitable for the sparse and uniformly distributed dataset, necessitating the development of a human-in-the-loop informed ensemble-based outlier detection (HIE-OD) method. The preprocessing steps involved using imputation techniques to retain four experiments and remove two highly correlated features as part of dimensionality reduction Additionally, 11 outliers were identified and removed, resulting in a refined dataset with 16 features and 139 observations.

2.3 Feature Importance Determination & Feature Selection

To effectively handle sparse data, as encountered in this study, feature selection is crucial. Traditional grid-search methods are unsuitable for exploring such complex input spaces due to a high risk of converging to local minima, especially in sparse datasets. Therefore, an evolutionary, multi-objective feature selection approach is recommended. These methods not only provide a comprehensive exploration of complex input spaces but also act as a form of regularization. However, a significant challenge with these methods is the instability of feature selection in high-dimensional spaces with limited observations, which can lead to poor predictive performance [5, 6].

To address these issues, we used the Informed Non-Dominated Sorting Genetic Algorithm II (I-NSGA-II) as a multi-objective feature selector. This approach effectively tackles both challenges associated with evolutionary multi-objective feature selection. The I-NSGA-II algorithm incorporates the most important features, as determined by the prior ensemble-based feature importance determination [2], as part of its initial solution. It then searches for additional features that are relevant to predicting the mechanical properties of UHPC, thereby improving both stability and predictive accuracy. The performance of the proposed I-NSGA-II was compared to the traditional NSGA-II [7] using ten different machine learning algorithms [8].

2.4 UHPC Manufacturing Process Modeling

The UHPC manufacturing process was modeled and evaluated using the leave-one-out cross-validation technique in the final step. This followed the generation of datapoints in the *Design of Experiments* step, the cleaning of data in the *Data Preprocessing* step, and the selection of the most important features and the best algorithm in the *Feature and Algorithm Selection* step.

3 Results and Future Work

The proposed framework was validated on two experimental datasets, proving its effectiveness in addressing challenges related to data dimensionality and limited sample size. Future work will focus on advancing the modeling phase by incorporating more complex and innovative algorithms, as well as testing the framework with additional benchmark functions.

Acknowledgment

This project (HA project no. 1011/21-13) is financed with funds of LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence).

LOEWE
Exzellente Forschung für
Hessens Zukunft

HESSEN
Hessisches Ministerium
für Wissenschaft und Kunst

References

[1] A. Abrishambaf, M. Pimentel, S. Nunes and C. Costa. “Multi-level study on UHPFRC incorporating ECat”. *Construction and Building Materials* 318. p. 125976. 2022.

[2] F. Rezazadeh P., A. Dürrbaum, G. Zimmermann and A. Kroll. “Leveraging ensemble structures to elucidate the impact of factors that influence the quality of ultra–high performance concrete”. In: *IEEE Symposium Series on Computational Intelligence (SSCI)* Mexico City. pp. 180–187. 2023.

[3] G. Taguchi. “System of experimental design: engineering methods to optimize quality and minimize costs”. *UNIPUB/Kraus International Publications, American Supplier Institute* 1987.

[4] M. Stein. “Large sample properties of simulations using Latin hypercube sampling”. *Technometrics* 29.2. pp. 143–151. 1987.

[5] M. Amoozegar and B. Minaei-Bidgoli. “Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism”. *Expert Systems with Applications* 113. pp. 499–514. 2018.

[6] R. Jiao, B.H. Nguyen, B. Xue and M. Zhang. “A survey on evolutionary multiobjective feature selection in classification: approaches, applications, and challenges”. *IEEE Transactions on Evolutionary Computation* p. 1. 2023.

[7] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. *IEEE Transactions on Evolutionary Computation* 6.2. pp. 182–197. 2002.

[8] F. Rezazadeh and A. Kroll. “Predicting the compressive strength of concrete up to 28 days-ahead: Comparison of 16 machine learning algorithms on benchmark datasets”. In: *Proceedings - 32. Workshop Computational Intelligence* Berlin. pp. 53–75. 2022.

A Novel Ranking Scheme for the Performance Analysis of Stochastic Optimization Algorithms using the Principles of Severity

Sowmya Chandrasekaran, Thomas Bartz-Beielstein

Institute for Data Science, Engineering, and Analytics, TH Köln, Steinmüllerallee 1, 51643 Gummersbach, Germany
E-Mail: {sowmya.chandrasekaran,thomas.bartz-beielstein}@th-koeln.de).

Abstract

Stochastic optimization algorithms have been successfully applied in several domains to find optimal solutions. Because of the ever-growing complexity of integrated systems, novel stochastic algorithms are being proposed, which makes the task of the performance analysis of the algorithms extremely important. This paper provides a novel ranking scheme to rank the algorithms over multiple single-objective optimization problems. The results of the algorithms are compared using a robust bootstrapping-based hypothesis testing procedure that is based on the principles of severity. Analogous to the football league scoring scheme, we propose pairwise comparison of algorithms as in league competition. Each algorithm accumulates points and a performance metric of how good or bad it performed against other algorithms analogous to the goal differences metric in the football league scoring system. The goal differences performance metric can be used not only as a tie-breaker but also to obtain a quantitative performance of each algorithm. The key novelty of the proposed ranking scheme is that it takes into account the performance of each algorithm considering the magnitude of the achieved performance improvement along with its practical relevance and does not have any distributional assumptions. To demonstrate the advantages of the proposed ranking scheme, we compare the expected run-time metrics of three hyperparameter optimization (HPO) procedures, namely, Irace, a mixed-integer parallel efficient global optimization (MIP-EGO), the mixed-integer evolution strategy (MIES), along with (1+1)EA and grid search(GS) on a genetic algorithm framework for Pseudo-Boolean Optimization (PBO) Suite of 25 problems. The proposed ranking scheme is compared to classical hypothesis testing and the analysis of the results shows that the results are comparable and our proposed ranking showcases many additional benefits.

1 Introduction

Numerous new meta-heuristic algorithms are being proposed to solve various complex problems. This makes the analysis of the performances of the algorithms to a relevant set of problems an inevitable task. Generally, the performances of the stochastic optimization algorithms are evaluated based on solution quality or utilized budget [1]. Here, the solution quality measures how close the solution obtained by an algorithm is with respect to the global optimum or the best-known value. This is referred to as the *fixed-budget* measure, where the achievable solution quality for a fixed budget is obtained. In the *fixed-target* perspective, the time required by an algorithm to hit the desired solution quality is measured. The time required can be CPU time or function evaluations. Typically, the CPU time can be dependent on many factors like computing environment, hardware resources, etc. Hence, the Function Evaluations (FE) is considered as an alternative time measure, where the number of times the objective function evaluated is measured. To test the robustness of the algorithm's performances, the algorithms are tested under uncertainty, noise, etc. The scalability measures the ability of the algorithm as the dimension of the problem increases. Be it the fixed-target or fixed-budget measure, due to the stochastic nature of the algorithm, there exists randomness in the performance of algorithms. Executing the same algorithm repeatedly can produce different solutions for the same inputs. Hence, there is a need for rigorous analysis of the performances of stochastic optimization algorithms.

In recent years, descriptive analysis (e.g. mean, median, best, worst and standard deviation) has turned out to be necessary but not sufficient metrics in analysing the performances of the algorithms. Statistical analysis plays a crucial role in comparing the performances of the algorithms [1]. A commonly used statistical tool over several years is hypothesis testing [15]. In order to compare the performances of algorithms, the null hypothesis can be formulated as *There is no statistically significant performance difference between a pair of algorithms vs* the alternative hypothesis as *There exists a statistically significant performance difference between a pair of algorithms.* Hypothesis testing can be broadly classified into parametric and non-parametric tests. While the former assumes a specific type of probability distribution of the data and makes inferences about the parameters of the distribution, the latter do not make any explicit assumptions about the data or its underlying distributions. The non-parametric tests are used when the assumptions for the safe use of parametric tests are not met. In both procedures, the p-value be the measure for deciding whether to retain or reject the null hypothesis. Since the statistical significance is decided in the form of a yes or no fashion that is based only on the p-value, the hypothesis testing procedure is criticized as black and white thinking [3]. Considering these criticisms, in [20], the American Statistical Association (ASA) explains the scope of p-value, wherein it emphasizes on considering additional appropriate measures along with p-value for a scientific decision. In [7], this issue is addressed using a measure, severity, which is a form of attained power [14]. More precisely, the concept of using severity as a tool for single pairwise comparison is proposed in [7], where both the statistical significance and also the practical relevance of the algorithms performances are measured. Also, the concept of severity is utilized for the analysis of the performances of hyper-parameter tuning in machine and deep learning algorithms [2].

The purpose of this paper is to propose a novel ranking scheme, that ranks the algorithms in a robust statistical fashion based on their performances considering the statistical significance, practical significance and magnitude of the achieved performance improvement. The resulting ranking scheme is analogous to the football league ranking system which has both the *points* scored and goal differences metric. Considering Multiple Algorithm Multiple Problem design (MAMP), each pairwise comparison of algorithms is treated similarly to a football game between two teams in a football league competition. The outcome of each game can be a *win*, a *draw*, or a *loss* for each algorithm on each problem based on which the *points* and *goal difference* are obtained. To the best of our knowledge, this may be the first statistical ranking scheme, which takes into account the win or loss of an algorithm along with the magnitude of the corresponding win or loss, i.e, in terms of the positive or negative goal differences. At the end of the league competitions, the algorithms are ranked based on the *points* and *goal difference*(GD).

In [7], the severity for various values of discrepancies is evaluated. The choice of evaluating varied discrepancies can be a little time-consuming. To overcome this issue, in this proposed paper, we have introduced a simple yet very effective concept, where the user needs to choose the desired severity, $S \in [0,1]$ (same as desired power). Note, this is the same as the desired power and not an explicit additional parameter. Then, the supported discrepancy for the chosen severity is obtained as an outcome. Based on outcome each algorithm attains two scoring as points and goal differences.

The paper is organized as follows: Section 2 explains the existing ranking frameworks to evaluate the performances of the optimization algorithms. Section 3.1 explains the concept of severity. Section 3.2 summarizes the proposed football based ranking scheme. In Section 4, a specific set of hyperparameter optimization techniques are evaluated for a family of genetic algorithms and are tested using the PBO Suite of 25 problems. Section 5 concludes with a summary and outlook.
Reproducibility: The source code is made available at `https://github.com/sowmyachandrasekaran17/algRanking`. The data and results of the experiment are available in [6].

2 Related Works

There exist some ranking schemes in the literature which were proposed to compare the performances of the optimization algorithms as in [5,9,10,16,18]. In [18], an empirical chess rating system for evolutionary algorithms using Glicko-2 rating is proposed. Here, the evolutionary algorithms are treated as chess players, and a pairwise comparison of two algorithms is considered as one game. Each game outcome can be a *win*, a *draw*, or a *loss*. At the end of the tournament, each algorithm is represented by rating(R), rating deviation (RD), and rating volatility(σ). Despite not being statistically analyzed, this rating system suffers from other issues. Firstly, the ordering of the games affects the final rating, though it is randomly selected. Furthermore, the overlapping of the confidence intervals might lead to statistical inconsistency as explained in [9]. Finally, the magnitude of the win or loss is not considered in the rating system.

More recently, different variants of deep statistic-based comparison (DSC) tool have been proposed [9, 10]. The advantage of this [10] ranking scheme is that it is based on the whole distribution rather than comparing mean or medians. In MAMP design, multiple pairwise comparisons of algorithms are performed using the non-parametric Kolmogorov-Smirnov or Anderson-Darling test and only the p-values determine the win or loss. Though the practical significance is addressed in DSC, the magnitude of the win or loss is not considered. Also, since the practical significance is directly included in the hypothesis formulation, the approach can be more conservative.

In [5, 16], the statistical comparison of the performances of the evolutionary algorithms is performed using Bayesian inferences. Though the identification of prior probabilities is an issue [11], the ranking of the algorithms is only based on the Bayesian probability of an algorithm being the best performer. The magnitude of the performance differences cannot be obtained.

3 Proposed Ranking Scheme

3.1 Concept of Severity

In order to explain the concept of hypothesis testing and severity, let us assume Normal, Independent, and Identically Distributed (NIID) data[1]. Let us consider algorithm A, say $\mathbf{a} = (a_1, \ldots, a_n)$, representing the function evaluations required by Algorithm A to achieve a specific target solution for n runs. Similarly,

---

[1] In the proposed ranking scheme, we do not assume the normality of the data. Here it is assumed to simplify the explanation of the concept and without the loss of generality, the concept can be adapted to the cases where the distribution is not known.

algorithm B, say $\mathbf{b} = (b_1, \ldots, b_n)$, represents the function evaluations required by Algorithm B to achieve the same target solution for n runs. We evaluate the performance of the algorithms repeatedly for n runs to handle the randomness in the evaluation metric and to obtain a reliable estimate of the metric.

The hypothesis testing is performed as an upper tail test of the mean differences as it is an optimization minimization problem. The difference vector $\mathbf{x}$ can be defined as $\mathbf{x} = (x_1, \ldots, x_n)$, where $x_i = a_i - b_i, \forall i = \{1, \ldots, n\}$ and $\bar{\mathbf{x}}$ denote the mean of the vector $\mathbf{x}$.
H_0 : B does not achieve less FE than $A \implies \bar{\mathbf{x}} \leq 0 \implies$ Loss for B.
H_1 : B achieves less FE than $A \implies \bar{\mathbf{x}} > 0 \implies$ Win for B.

$$H_0 : \bar{\mathbf{x}} \leq 0; \text{vs. } H_1 : \bar{\mathbf{x}} > 0,$$

$$decision = \begin{cases} \text{not-Reject} H_0, & \text{if } d(\mathbf{X}) \leq u_{1-\alpha}, \\ \text{Reject} H_0, & \text{otherwise}, \end{cases}$$

where $u_{1-\alpha}$ is the upper tail cut-off point of the normal distribution, which cuts the upper-tail probability of α, and the test statistic $d(\mathbf{X})$ can be represented as

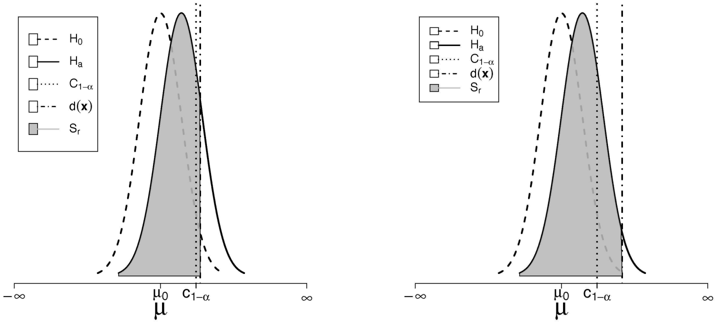
$$d(\mathbf{X}) = \frac{\bar{X} - \mu_0}{\sigma_x},$$

where standard error $\sigma_x = \frac{\sigma}{\sqrt{N}}$ and μ_0 is the hypothesized mean under H_0. If a test statistic is observed beyond the cut-off point, we reject the H_0 at a significance level α. Here, the values for α, β, and hence power $(1-\beta)$ are pre specified before the experiment is performed.

In the context of ranking the algorithms based on the results of parametric or non-parametric hypothesis testing, based only on the p-value, i.e, if a test statistic is observed beyond the cut-off point, the H_0 is rejected and hence B wins and is declared of achieving less FE than A. Also, B gains a point just based on this decision. This is criticized as black and white thinking.
Severity, a form of attained power [14], is a probability analogous to the p-value under the alternative hypothesis rather than one under the null. In case of win or loss, the magnitude of the performance improvement is measured in terms of severity as S_r and S_{nr} respectively. The loss S_{nr} values increase monotonically

from 0 to 1. The won S_r values decrease monotonically from 1 to 0. The closer the value is to 1, the more reliable the decision made with the hypothesis test. Generally, a severity value of 0.8 is considered reliable support. The differences between α, p-value, power, and severity is provided in Table 1 in [7].

The importance of severity representation of won is shown in Figure 1. In a similar fashion, the severity representation of loss can also be visualized. In Figure 1, only one value for the alternate hypothesis is visualized. However, in practice, we evaluate the severity for specific possible values under the H_1. The different values under the alternative hypothesis for which the compatibility is assessed will henceforth be called discrepancy, δ. It measures how discrepant is the performance improvement when compared to the null improvement. In the context of algorithm ranking, since we consider the FE as the metric, we can evaluate if B won over A with 1000 FE or 10 FE, thereby quantization the magnitude of the victory.



(a) Won Scenario 1 :Severity to Reject H0

(b) Won Scenario 2 :Severity to Reject H0

Figure 1: Illustration of two scenarios of S_r under the alternate hypothesis. In both cases, the actual test statistic $d(\mathbf{x})$, falls outside the $u_{1-\alpha}$, the decision is to reject the null. The S_r is the area under the H_1 that is within the $d(\mathbf{x})$ (shaded area). Though in both cases, the decision is the same, severity sheds light in understanding the actual attained power of the test. In (a), less support for the decision won (shaded area) as $d(\mathbf{x})$ is closer to the cut-off point and in (b), more support for the won (shaded area) as $d(\mathbf{x})$ is way more from the cut-off point.

3.2 Proposed Algorithm

The football league scoring system has been designed to provide a fair and objective method for ranking teams based on their performances. Each football match between two teams, has three possible outcomes: win, loss or draw. Based on the outcome the teams get *points*, 3 for winning, 1 for drawing and 0 for losing. The *points* earned by each team from matches played over a season are cumulatively added and the final rankings for the season is obtained. In addition, the goal difference, which is the difference between the number of goals scored minus the number of goals conceded in matches is calculated. Goal difference serves as a tiebreaker if two or more teams have the same *points*.

Analogous to the football league scoring scheme, we propose a novel ranking scheme for ordering the performances of several stochastic algorithms on a set of well-known bench-marking test functions ($\mathcal{F}$) as Algorithm 1. Let $\mathcal{A} := \{A_i, \forall i \in \{1,\ldots,k\}\}$ denote the set of all algorithms that must be ranked and $\mathcal{F} := \{F_l, \forall l \in \{1,\ldots,m\}\}$ denote the set of all test functions. An experiment is performed where each algorithm $A_i \in \mathcal{A}$ is evaluated on each of the test functions $F_l \in \mathcal{F}$ for n runs. Let $\mathcal{C}$ represent set of all possible pair-wise algorithm comparisons, $\mathcal{Y}$ represent the set of all corresponding solutions and are defined as

$$\mathcal{C} := \left\{(A_i, A_j)_l, \forall i, j \in \{1,\ldots,k\}, i \neq j, \forall l \in \{1,\ldots,m\}\right\}, \tag{1}$$

$$\mathcal{Y} := \left\{(\mathbf{y}_i, \mathbf{y}_j)_l, \forall i, j \in \{1,\ldots,k\}, i \neq j, \forall l \in \{1,\ldots,m\}\right\}, \tag{2}$$

where $\mathbf{y}_i \in \mathbb{R}^n$ denotes the solution of i^{th} algorithm A_i for n runs for a given function in $\mathcal{F}$. The ranking scheme requires each pairwise comparison defined in $\mathcal{C}$ and set $\mathcal{Y}$ should be obtained and this results in a total of $k \times k - 1 \times m$ comparisons. For each $(A_i, A_j)_l \in \mathcal{C}$, bootstrapping-based t-test is performed. Sampling with replacement is done to attain a better estimate of the metric. This procedure also eliminates the dependence of the outcome on the ordering of the optimization runs. For a given value of $i, j \in \{1,\ldots,k\}$, $l \in \{1,\ldots,m\}, i \neq j$, the following steps are performed.

Algorithm 1.: Proposed Ranking Scheme

```
for each pair-wise comparison $(A_i,A_j)_l \in \mathcal{C}$ do
    Require: $(\mathbf{y}_i,\mathbf{y}_j)_l = ((y_i^1,\dots,y_i^n),(y_j^1,\dots,y_j^n))_l$, $\alpha$, $S$,$n_b$, $\delta_p$
    Formulate Hypothesis $H_0 : \bar{\mathbf{x}} \le 0$ vs. $H_1 : \bar{\mathbf{x}} > 0$
    Evaluate observed sample mean difference $t_{obs} = \overline{\mathbf{y}_i} - \overline{\mathbf{y}_j}$
    Combine $\mathbf{I} = \widehat{\mathbf{y}_i\mathbf{y}_j}$
    repeat
        Draw a bootstrap sample of $2n$ observations with replacement from $\mathbf{I}$
        Let the mean of the first $n$ observation be $\overline{\mathbf{y}_i}^*$
        Let the last $n$ observations be $\overline{\mathbf{y}_j}^*$
        Evaluate $t^{*\mathrm{bs}} = \overline{\mathbf{y}_i}^* - \overline{\mathbf{y}_j}^*$
        Evaluate $t_s^{*\mathrm{bs}} = \overline{\mathbf{y}_i}^* - \overline{\mathbf{y}_j}^* - \delta$
    until $n_b$ times
    Calculate $p \approx \frac{\#(t^{*\mathrm{bs}} \ge t_{obs})}{n_b}$
    Obtain $adjusted-p$ based on BH correction
    if $adjusted-p \le \alpha$ then
        decision: Reject $H_0$ and $\delta^* = \min_\delta S - \frac{\#(t_s^{*\mathrm{bs}} \le t_{obs})}{n_b}$
        if $\delta \le \delta_p$ then
            $points = 1$ and $GD = 0$
        else
            $points = 3$ and $GD = \lfloor \frac{\delta^*}{\delta_p} \rfloor$
    if $adjusted-p > \alpha$ then
        decision: not-Reject $H_0$ and $\delta^* = \max_\delta S - \frac{\#(t_s^{*\mathrm{bs}} > t_{obs})}{n_b}$
        $points = 0$ and $GD = \lfloor \frac{\delta^*}{\delta_p} \rfloor$
Obtain cumulative $points$ and $GD$
Rank based on $points$ and $GD$
```

1. Merge the results of the algorithms to obtain $\mathbf{I} := \widehat{\mathbf{y}_i\mathbf{y}_j}$ of size $2n$ and compute the observed sample mean difference $t_{obs} := \bar{\mathbf{y}}_i - \bar{\mathbf{y}}_j$.
2. Draw a bootstrap sample of $2n$ observations with replacement from $\mathbf{I}$ and evaluate the bootstrap test statistic $t^{*\mathrm{bs}} = \bar{\mathbf{y}}_i^* - \bar{\mathbf{y}}_j^*$.
3. Repeat the procedure based on the bootstrapping re-sample size n_b, and estimate p-value as the number of times the bootstrap test statistic $t^{*\mathrm{bs}}$ was found to be greater than the t_{obs} in n_b samples. I.e., $p \approx \frac{\#(t^{*\mathrm{bs}} \ge t_{obs})}{n_b}$

4. Adjust the p-value using Benjamini-Hochberg (BH) correction [4].
5. When the adjusted p-value is found to be significant, i.e., less than the specified α, then reject H_0, else do not reject H_0. Based on the decision and the chosen severity requirement, S, obtain the supported δ.

The significance level α, the desired severity, $S \in [0, 1]$, (same as desired power) and the practically relevant δ_p should be chosen by the user. As the power of the test is usually chosen as 80 percent or 95 percent based on the problem domain, similarly, the recommended severity is chosen at 80 percent or 95 percent and α of 95 percent. Based on the performance metric and the problem domain, to identify if the achieved performance improvement is better than the practical relevance, the δ_p shall be chosen carefully. In case of FE as the metric where the available budget is 100000 FEs, minimum of 100 FE improvement at desired severity can be considered as a practically relevant improvement. In case of CPU time as the metric, depending on the application, meaningful time can be chosen. E.g., for an optimization algorithm implemented in an autonomous car, this value can be in centiseconds, and for offline scheduling problems, it can be in several minutes to hours. The supported δ obtained from the algorithm gives a measure of change in statistics required to achieve the expected severity. The rounded-down value of the ratio of δ and δ_p provides the goal difference metric in each match, thereby quantifying the size of a win or a loss.

The *points* scored for each algorithm are similar to the football league scoring system and are obtained based on the decision of the bootstrapped hypothesis testing, the statistical significance of the result, i.e. supported δ at desired severity and the practical relevance of the win or loss, whether, supported $\delta < \delta_p$ or $\delta > \delta_p$. The *goal difference* measures how much the performance is better/worse in terms of practical relevance. The *points* and *goal difference* are calculated as

$$Outcome = \begin{cases} points\text{=3, GD=}\lfloor \delta/\delta_p \rfloor > 0, \text{ if Reject} H_0 \text{ and} & \text{if } \delta > \delta_p, \\ points\text{=1, GD=}\lfloor \delta/\delta_p \rfloor = 0, \text{ if Reject} H_0 \text{ and} & \text{if } \delta < \delta_p, \\ points\text{=0, GD=}\lfloor \delta/\delta_p \rfloor \leq 0, \text{ if not-Reject} H_0. \end{cases}$$

Upon completion of all pairwise comparisons in $\mathcal{C}$, the cumulative *points* and the cumulative goal differences are calculated and the algorithms are ranked based on the *points*. In addition, the mean and standard deviations of the *points* for each algorithms among all functions can be obtained. The points awarded to algorithms that performed well with practical significance are weighted more than twice (three times to be precise) the points awarded to algorithms that performed only statistically significant. This takes into account the fact that practical significance is more relevant in real-world applications and therefore provides a better estimate of the overall performance of the algorithms. On the other hand, weighting practical significance more than three times may skew the results more towards practical significance and therefore statistical significance will have less influence on the final result.

3.3 Properties of the Ranking Scheme

The proposed ranking scheme demonstrates a weak order since it satisfies the comparability, reflexivity, transitivity, anti symmetry properties and it allows for ties, if goal difference does not differentiate the algorithms [2]. Let us define a relation $R(A_i,A_j)$ such that for any two algorithms A_i and A_j in $\mathcal{A}$, P_i, P_j denote the points and GD_i, GD_j denote the goal difference of algorithm A_i and A_j respectively. Note that the symbols $>,<,=$ represent that one algorithm is ranked higher than, lower than, equal to the other algorithm respectively. Then relation $R(A_i,A_j)$ between two algorithms can be formally defined as

$$R(A_i,A_j) = \begin{cases} A_i > A_j, \text{if } (\ P_i > P_j) \text{or } (P_i = P_j \text{ and } GD_i > GD_j), \\ A_j > A_i, \text{if } (P_j > P_i) \text{ or } (P_j = P_i \text{ and } GD_j > GD_i), \\ A_i = A_j, \text{ if } P_i = P_j \text{ and } GD_i = GD_j. \end{cases}$$

Comparability: Every pair of algorithms $(A_i,A_j)_l \in \mathcal{C}$ are comparable by the definition of $R(A_i,A_j)$

Reflexivity: For every algorithm compared, it is equivalent to itself. *Outcome* gives unique value associated with each algorithm. For any $A_i \in \mathcal{A}$,

$$A_i = A_i, \text{ since } P_i = P_i \text{ and } GD_i = GD_i$$

Transitivity: Since the ranking scheme is based on total *Outcome*, for any $A_i, A_j, A_k \in \mathcal{A}$, $A_i > A_j$ and $A_j > A_k$, then $A_i > A_k$ because of the following.

$$\text{if } P_i > P_j \text{ and } P_j > P_k\text{, then } P_i > P_k$$

$$\text{if } (P_i = P_j \text{ and } GD_i > GD_j) \text{ and } (P_j = P_k \text{ and } GD_j > GD_k)$$

$$\text{then } P_i = P_k \text{ and } GD_i > GD_k.$$

Anti symmetry: The anti-symmetry property is defined by the following relationship.

$$\text{if } A_i > A_j \text{ and } A_j > A_i \implies A_i \simeq A_j$$

This can be proven directly from the definition of $R(A_i, A_j)$ because the condition can be satisfied only when the algorithms are ranked equal.

4 Case study

We compare the expected run time metrics of three hyper parameter optimization techniques along with (1+1)EA and grid search for a family of genetic algorithms on PBO Suite of 25 problems obtained from [21]. The 25 PBO functions include from onemax, leadingones, a linear function with harmonic weights, various W-model-transformes of onemax and leadingones, low autocorrelation binary sequences, ising models, maximum independent vertex set, N-queens problems. The compared HPO techniques include Irace [13], a mixed-integer parallel efficient global optimization [17], the mixed-integer evolution strategy [12].

As (1+1)EA has shown good performance for PBO in [8] it is considered a baseline. The goal is analysing the impact of mutation, crossover, and its combination on a family of $(\mu + \lambda)$ GA algorithms, which results in four tuning parameters: Parent population size, $\mu \in [100]$, Offspring population size ,$\lambda \in [100]$, mutation rate, $P_m \in [.005, .5]$, cross over probability $P_c \in [0, 1]$. Each of the HPO techniques is allocated a budget of 5000 target runs, where each target run refers to 10 independent runs of the $(\mu + \lambda)$ GA configuration suggestion by the HPO techniques. Two different performance metrics are considered, namely, minimizing the expected runtime (ERT) and maximizing the Area under the empirical CDF curve of running times (AUC). This results in 9 algorithms to be compared: (1+1)EA, GS.AUC, GS.ERT, Irace.AUC, Irace.ERT, MIES.AUC, MIES.ERT, MIP.EGO.AUC, MIP.EGO.ERT. The $(\mu+\lambda)$ GA configurations provided by each of the HPO technique are evaluated for a budget of 50000 function evaluations and the ERT values are obtained for the PBO problems with respect to the targets defined in Table 1 in [21]. And for AUC, set of 100 equally spaced targets ranging from 0 to the targets defined in Table 1 in [21] is considered. Upon identification of the best tuning parameters by each HPO technique, 100 independent runs for these tuned settings are performed and used for further analysis. For each of the algorithm, results of these 100 runs at desired target defined in Table 1 in [21] is used as performance data of this case study and is obtained using IOHanalyzer [19] as explained in [21]. In all runs, values are capped at the budget 50000 function evaluations if the algorithm cannot find the target. The experimental setup for our ranking scheme is as follows: the significance level α is chosen as 0.05, and the desired severity is at recommended value of 0.8. Since the function evaluation is the performance metric in our fixed-target perspective and the total allocated budget is 50000 function evaluations, the practically significant performance improvement, δ_p, can be chosen as minimum of 500 function evaluations. The re-sample size n_b is chosen as 10000. The classical bootstrapped-based hypothesis testing (HT) is also performed for same α and n_b. Each algorithm scores one point for the decision RejectH_0 or zero otherwise. The resulting ranking results are shown in Table 1.

Without going into the details of the performance of the algorithms for each function, we can understand the quantitative performance of each algorithm from the table provided. For example, (1+1)EA clearly outperforms all the other algorithms and tops the table. However, looking at the GD metric, it can be observed that (1+1)EA has performed really poorly for the cases where it failed to win leading to a big negative goal difference at the end. Though MIES.ERT, MIES.AUC, Irace.AUC lags behind (1+1)EA in the *points*, they have not performed really poorly for the cases it lost against other algorithms. If one would like to minimize the worst-case scenarios to achieve robustness, MIES.ERT can be chosen to be applied instead of (1+1)EA. In this case, we may not achieve the optimum the fastest. However, the results will not be the slowest for some classes of problems either.

Also, our scheme does not suffer from the problem of aggregating results in ranking as function-wise ranking metrics for each algorithm are analyzed as explained in Figure 2. The (1+1)EA algorithm was outperforming MIES.ERT for functions 6,7, 15-17, 19-21 and 23 as shown in Figure 2. However, the positive GD in these functions was very low. For functions 8-10,14,18, and 24, where the (1+1)EA could not outperform, the GD was negative. The MIES.ERT was able to outperform (1+1)EA in functions 2,3,8-14,18,24, and 25 with positive GD for the majority of the functions. The worst GD for MIES.ERT is approx. -100 for function 6 compared to the value of approx. -580 for function 14 in the case of (1+1)EA. This helps us make informed decisions for a given application. A complete picture of which algorithm was best/worst for which set of functions is made available, which is a major strength of the scheme. Considering page restrictions, function-wise ranking metrics is discussed only for the top 2 algorithms. However, the results for all other algorithms are made available online in [6].

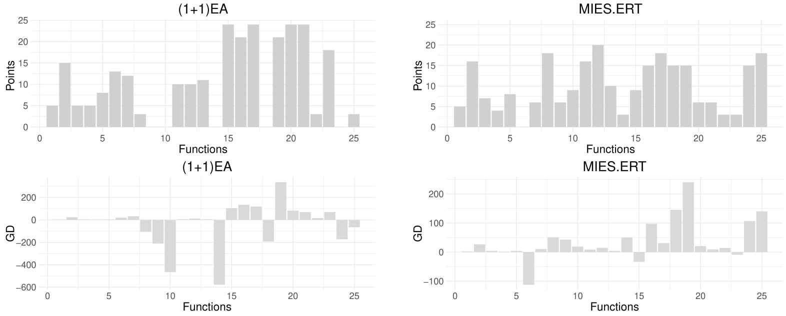


Figure 2: Function-wise ranking metrics of (1+1)EA and MIES-ERT Algorithms

Considering the average mean, median and SD statistics of the *points*, MIES.ERT, MIES.AUC and Irace.AUC exhibits similar performances. Also, the performances of Irace.ERT, MIP.EGO.ERT and MIP.EGO.AUC are comparable. The ranking produced by the classical boostrapped based HT is the same with the only exception of MIP.EGO.AUC outperforming MIP.EGO.ERT by one point. It is also evident that the performances of both algorithms in our proposed ranking scheme are comparable. Again, this highlights the importance of *GD* metric, which clearly showcases a very high positive goal difference for MIP.EGO.ERT and very large negative goal difference for MIP.EGO.AUC. This explains the order that is provided by the proposed algorithm which gives a larger weight for practical significance (i.e., 3 *points*).

Table 1: Proposed Ranking Scheme Results(S=0.8,δ_p=500) vs classical bootstrapped-based HT. The *points* and GD obtained are the cumulative *points* and GD obtained by each algorithm for all the 25 PBO problems. The position change (PC) in ranking positions is indicated for classical HT with (↑, ↓).

Algorithm	Proposed Ranking		classical HT	
	points	Goal Difference	*points*	PC
1+1 EA	259	-755	107	-
MIES.ERT	251	889	103	-
MIES.AUC	240	899	100	-
Irace.AUC	227	804	89	-
Irace.ERT	169	-207	67	-
MIP.EGO.ERT	161	368	59	↓ 1
MIP.EGO.AUC	144	-788	60	↑ 1
GS.ERT	113	-544	41	-
GS.AUC	91	-820	35	-

4.1 Sensitivity analysis of δ_p and S

The sensitivity of input parameters on the ranking solutions are validated at desired severity levels of $S = 0.5, 0.65, 0.8, 0.95$. Similarly, the practically relevant function evaluation is evaluated for a very wide range of $\delta_p = 50, 100, 250$, and the results of the ranking scheme for the resulting 16 experiments are compared. Table 2 presents the influence of severity on the algorithm ranking for δ_p of 500. As expected, as the severity values increase from 50% to 95%, the test becomes more stringent and this is evident in the decreasing trend of *points*. The order of the algorithm rankings is consistent even for various values of severity. Note, however the magnitude of the change is a function of statistics of the solution of the algorithms and cannot be adjudged prior to the experiments. Considering page limitations the detailed results of the influence of δ_p parameter

Table 2: Sensitivity analysis of parameter severity in Ranking Results (δ_p=500). Given a δ_p, the *points* achieved by each algorithm decreases monotonically with respect to the increase in severity. As severity increases, the algorithms that pass a more stringent hypothesis test decreases and accordingly scores less number of *points*. However, no prior statement can be given for the trend in GD for this case because it is a function of δ_p. It is important to note that the sequence of the rankings of the algorithms remains unaltered with various severity levels.

Algorithm	S 50%		S 65%		S 95%	
	points	GD	*points*	GD	*points*	GD
1+1 EA	271	-855	263	-806	249	-669
MIES.ERT	257	952	251	926	241	806
MIES.AUC	246	949	242	930	232	836
Irace.AUC	229	864	227	834	223	736
Irace.ERT	171	-205	171	-205	167	-224
MIP.EGO.ERT	165	344	161	355	159	375
MIP.EGO.AUC	148	-934	146	-865	142	-673
GS.ERT	113	-608	113	-578	109	-494
GS.AUC	95	-975	93	-903	91	-693

on the outcome of the rankings is published in [6]. The order of the algorithm rankings remains consistent for δ_p ranging from 100 to 500. Only when δ_p is 50, the MIP.EGO.AUC secures 178 *points* with -7877 GD and is ahead of MIP.EGO.ERT which secured 177 *points* with 3685 GD. However, this is not a significant improvement for MIP.EGO.AUC, which is evident with the highly negative goal differences. It is to be noted that choosing extreme values for δ_p might influence the sequence of ranking and that is intentional. However, the comparison for all algorithms is performed using the same δ_p and the relative performance improvement is obtained. Even in the existing ranking schemes, the practically relevant improvement parameter is defined as a user-defined variable [9, 10, 18], and extreme choice of this parameter will alter the results. Considering the page restrictions, only certain experiments are presented in Table 2. However, the resulting order of the rankings remained unaltered for the other set of experiments as published in [6]. In addition, repeating the same experiment for several times always produced the same results, convincing of the robust and stable outcome of the proposed ranking scheme. The CRS4EAs ranking scheme as proposed in [18], is used to compare the results obtained by our proposed ranking scheme. The results of the same 9 algorithms for 25

Table 3: Results of CRS4EAs ranking scheme for 2 random runs but for the same input settings. The algorithms are ranked as per the proposed ranking scheme and the position change (PC) in rankings based on CRS4EAs is indicated.

Algorithm	Trial 1				Trial 2			
	R	RD	σ	PC	R	RD	σ	PC
1+1 EA	1586	12.7	0.04	-	1578	12.7	0.04	↓1
MIES.ERT	1569	13.5	0.04	↓1	1574	13.8	0.04	↓1
MIES.AUC	1586	14.9	0.06	↑1	1604	14	0.05	↑2
Irace.AUC	1563	13.1	0.04	-	1559	14.7	0.05	-
Irace.ERT	1457	13.5	0.04	↓2	1451	12.7	0.04	↓2
MIP.EGO.ERT	1479	12.8	0.04	↑1	1495	13.5	0.04	↑1
MIP.EGO.AUC	1473	13.4	0.04	↑ 1	1480	13.1	0.04	↑ 1
GS.ERT	1377	14.2	0.05	↓ 1	1371	14.2	0.05	↓ 1
GS.AUC	1410	13.1	0.04	↑ 1	1400	13.9	0.04	↑ 1

PBO problems are provided as input and in total 5000 games were played. The implementation available in [19] is used for the experiments with the default suggested input parameter settings. The results are shown in Table 3. It is evident that for the same input, the resulting ranking and the ratings keep varying when the same experiment is repeated. Hence, consistent ranking is not observed even with the same inputs.

5 Summary and Outlook

A user-friendly novel ranking scheme based on football league system is proposed that takes into account the statistical significance, the practical significance, and the magnitude of the win or loss of the compared algorithms to determine the final ranking of the algorithms. The proposed scheme has the advantage that the order of comparison has no impact on the results along with there is no necessity of knowing the prior statistics of the compared algorithms. Since the practical significance is used separately from hypothesis testing, the resulting scheme does not make the test more conservative. The proposed scheme shows potential also for comparing machine learning, artificial intelligent(AI), and explainable AI algorithms. For didactic purposes, the HT is formulated considering the mean performance differences among algorithms. Nevertheless, the scheme can be used to compare the median and related performance measures as well.

References

[1] Bartz-Beielstein, T., Doerr, C., Bossek, J., Chandrasekaran, S., Eftimov, T., Fischbach, A., Kerschke, P., Lopez-Ibanez, M., Malan, K.M., Moore, J.H., et al.: Benchmarking in optimization: Best practice and open issues. arXiv preprint arXiv:2007.03488 (2020)

[2] Bartz-Beielstein, T., Mersmann, O., Chandrasekaran, S.: Ranking and result aggregation. In: Bartz, E., Bartz-Beielstein, T., Zaefferer, M., Mersmann, O. (eds.) Hyperparameter Tuning for Machine and Deep Learning with R: A Practical Guide, chap. 5, pp. 121–161. Springer Nature (2023)

[3] Benavoli, A., Corani, G., Demšar, J., Zaffalon, M.: Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. The Journal of Machine Learning Research **18**(1), 2653–2688 (2017)

[4] Benjamini, Y., Hochberg, Y.: Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society: Series B (Methodological) **57**(1), 289–300 (12 2018).

[5] Calvo, B., Shir, O.M., Ceberio, J., Doerr, C., Wang, H., Bäck, T., Lozano, J.A.: Bayesian performance analysis for black-box optimization benchmarking. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. pp. 1789–1797 (2019)

[6] Chandrasekaran, S.: Data and results of paper titled: A novel ranking scheme for the performance analysis of stochastic optimization algorithms using the principles of severity. `https://doi.org/10.5281/zenodo.10969073`

[7] Chandrasekaran, S., Bartz-Beielstein, T.: A robust statistical framework for the analysis of the performances of stochastic optimization algorithms using the principles of severity. In: International Conference on the Applications of Evolutionary Computation (Part of EvoStar). pp. 426–441. Springer (2023)

[8] Doerr, C., Ye, F., Horesh, N., Wang, H., Shir, O.M., Bäck, T.: Benchmarking discrete optimization heuristics with iohprofiler. Applied Soft Computing **88**, 106027 (2020)

[9] Eftimov, T., Korošec, P.: Identifying practical significance through statistical comparison of meta-heuristic stochastic optimization algorithms. Applied Soft Computing **85**, 105862 (2019)

[10] Eftimov, T., Korošec, P.: A novel statistical approach for comparing meta-heuristic stochastic optimization algorithms according to the distribution of solutions in the search space. Information Sciences **489**, 255–273 (2019)

[11] Gelman, A.: Objections to Bayesian statistics. Bayesian Analysis **3**(3), 445 – 449 (2008)

[12] Li, R., Emmerich, M.T., Eggermont, J., Bäck, T., Schütz, M., Dijkstra, J., Reiber, J.H.: Mixed integer evolution strategies for parameter optimization. Evolutionary computation **21**(1), 29–64 (2013)

[13] López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L.P., Birattari, M., Stützle, T.: The irace package: Iterated racing for automatic algorithm configuration. Operations Research Perspectives **3**, 43–58 (2016)

[14] Mayo, D.G., Spanos, A.: Severe testing as a basic concept in a neyman-pearson philosophy of induction. The British Journal for the Philosophy of Science **57**(2), 323–357 (2006)

[15] Neyman, J., Pearson, E.S.: On the use and interpretation of certain test criteria for purposes of statistical inference: Part i. Biometrika pp. 175–240 (1928)

[16] Rojas-Delgado, J., Ceberio, J., Calvo, B., Lozano, J.A.: Bayesian performance analysis for algorithm ranking comparison. IEEE Transactions on Evolutionary Computation **26**(6), 1281–1292 (2022)

[17] van Stein, B., Wang, H., Bäck, T.: Automatic configuration of deep neural networks with parallel efficient global optimization. In: 2019 International Joint Conference on Neural Networks (IJCNN). pp. 1–7. IEEE (2019)

[18] Veček, N., Mernik, M., Črepinšek, M.: A chess rating system for evolutionary algorithms: A new method for the comparison and ranking of evolutionary algorithms. Information Sciences **277**, 656–679 (2014)

[19] Wang, H., Vermetten, D., Ye, F., Doerr, C., Bäck, T.: Iohanalyzer: Detailed performance analyses for iterative optimization heuristics. ACM Trans. Evol. Learn. Optim. **2**(1) (apr 2022).

[20] Wasserstein, R.L., Lazar, N.A.: The ASA's statement on p-values: Context, process, and purpose. The American Statistician **70**(2), 129–133 (2016)

[21] Ye, F., Doerr, C., Wang, H., Bäck, T.: Automated configuration of genetic algorithms by tuning for anytime performance. IEEE Transactions on Evolutionary Computation **26**(6), 1526–1538 (2022)

EAP4EMSIG - Experiment Automation Pipeline for Event-Driven Microscopy to Smart Microfluidic Single-Cells Analysis

Nils Friederich[1,2,*], Angelo Jovin Yamachui Sitcheu[1,*],
Annika Nassal[1,2], Matthias Pesch[3], Erenus Yildiz[4],
Maximilian Beichter[1], Lukas Scholtes[4], Bahar Akbaba[1],
Thomas Lautenschlager[1], Oliver Neumann[1], Dietrich Kohlheyer[3],
Hanno Scharr[4], Johannes Seiffarth[3,5,#], Katharina Nöh[3,#],
Ralf Mikut[1,#]

[1] Institute for Automation and Applied Informatics (IAI)
[2] Institute of Biological and Chemical Systems (IBCS)
Karlsruhe Institute of Technology
[3] Institute of Bio- and Geosciences (IBG-1)
[4] Institute for Data Science and Machine Learning (IAS-8)
Forschungszentrum Jülich GmbH
[5] Computational Systems Biology (AVT-CSB)
RWTH Aachen University
*Contributed equally
#Supervised equally

Abstract

Microfluidic Live-Cell Imaging (MLCI) generates high-quality data that allows biotechnologists to study cellular growth dynamics in detail. However, obtaining these continuous data over extended periods is challenging, particularly in achieving accurate and consistent real-time event classification at the intersection of imaging and stochastic biology. To address this issue, we introduce the Experiment Automation Pipeline for Event-Driven Microscopy to Smart Microfluidic Single-Cells Analysis (EAP4EMSIG). In particular, we present initial zero-shot results from the real-time segmentation module of our approach. Our findings indicate that among four State-Of-The-Art (SOTA) segmentation methods evaluated, Omnipose delivers the highest Panoptic Quality (PQ) score of 0.9336, while Contour Proposal Network (CPN) achieves the fastest inference time of 185 ms with the second-highest PQ score of 0.8575. Furthermore, we observed that the vision foundation model Segment Anything is unsuitable for this particular use case.

1 Introduction

What are microbes? Microbes, also known as microorganisms, are a group of tiny living organisms that are invisible to the naked eye. This group includes bacteria, archaea, fungi and protists [5]. Microbes are present almost everywhere on Earth, from harsh environments such as hydrothermal vents to the human body, where they outnumber human cells by a factor of around 1.3 [54]. Despite their tiny size, microbes play crucial roles in various ecological and biological processes, making them essential for life on Earth [30].

Why are microbes relevant? Microbes are relevant for several reasons. The first is ecological balance, where microbes are essential in the nutrient cycle, decomposing organic matter and contributing to soil fertility [57]. They are crucial for the carbon, nitrogen and sulfur cycles that sustain life on Earth [27]. Second, in human health, the human microbiome aids digestion, produces essential vitamins and protects against pathogenic microbes [38]. Disruptions in the microbiome can lead to health issues such as infections, obesity and autoimmune diseases [7,29]. Finally, in the context of industrial applications, microbes are harnessed in biotechnology, pharmaceuticals and agriculture. They are used to produce antibiotics, biofuels and fermented foods [25]. Microbial enzymes are also crucial in many manufacturing processes [46].

Why is research on microbes essential? Research on microbes is important due to their impact on health, industry and the environment. Understanding microbial behavior, genetics and interactions can advance all three areas. In medical science, it is crucial to study pathogens to help develop vaccines and treatments for infectious diseases [2]. Microbe research can potentially reveal new therapies for chronic diseases [3]. In environmental protection, microbes can be used in bioremediation to clean up oil spills and toxic waste [18]. Therefore, understanding microbial ecosystems can help conservation efforts and combat climate change. In biotechnology, microbial research can lead to the development of new applications, such as using microbes to produce valuable compounds, e.g., insulin or biodegradable plastics [36].

Why is the segmentation of microbes relevant? While some biological analysis is possible at the macroscopic level, other results can only be obtained by studying organisms at the microscopic single-cell level. MLCI particularly enables an understanding of single-cell growth and growth heterogeneity due to very small volumes. For example, the effect of antibiotic concentrations on organism performance can be analyzed through such experiments. Understanding the dynamics of microbes at this single-cell level therefore requires accurate and precise automated cell segmentation, as large amounts of data acquired using automated microscopy must be analyzed to obtain relevant results. The segmented data can then be used to make statements about the organism's growth as a function of various other factors.

What is the challenge in microbe research? MLCI experiments with microbes are usually not carried out on a single colony but in parallel on thousands. To achieve this, the microfluidic device is infused with a cell suspension and cells are randomly seeded into the growth chambers, where they remain trapped. Optimally, a microbial colony grows in each chamber. In a standard growth experiment, seeded cells grow until the chamber is filled with densely packed cells, which can be 1000 ends, which marks the end of the experiment. Subsequent experiment examination requires analyzing thousands of colonies in parallel, with thousands of microbes in some cases. Each chamber must be manually assessed to determine whether it meets the experiment's objectives, even as some chambers become irrelevant as the experiment advances. This process is extremely time-consuming, costly, strenuous, monotonous and nearly impossible. Therefore, automated and intelligent processing, analysis and experiment planning are required.

How does this paper address this challenge? In this paper, we introduce the EAP4EMSIG, designed to automate and intelligently execute MLCI experiments, during which the human expert specifies settings, monitors progress and intervenes only to address any issues that may arise. We explain the concept of the pipeline and its eight primary modules. To achieve this, a literature review (see Section 2) and an extensive description (see Section 3) of each module are provided.

Since real-time data evaluation, inference and incorporation into the experimental design are central parts of our entire Experiment Automation Pipeline (EAP) pipeline, our work will compare initial results. We will compare the results related to the Average Precision (AP) [28] score, PQ [23] score and inference time of four SOTA methods from the task, domain and foundation areas, using a large publicly available microbial benchmark dataset [51, 52] (see Section 4). For this purpose, their zero-shot abilities and their real-time capability will be analyzed and investigated to determine which models are suitable for retraining. Additionally, we will evaluate whether using a foundation model potentially leads to better results through improved generalization.

2 Related Work

Experiment Automation Pipelines. Various EAP tools have been developed, ranging from local standalone projects [10] to cloud-based tools [34]. Some methods focus on automating the data analysis part [17, 37], others focus on automating the data acquisition part [43], particularly on microscope control [40, 42] and event-based image acquisition [6, 33]. However, very few generic tools for full experiment automation have been proposed due to the complexity of combining the experiments' software, hardware and biological components. One example is the PYthon Microscopy Environment (PYME)[1] open-source

---

[1] https://www.python-microscopy.org/

package, which offers data acquisition, processing, exploration and visualization modules. PYME is, however, only tailored for super-resolution techniques. Another example is Cheetah [39], a Python library that automates real-time cybergenetic experiments. It offers limited microscope control capabilities and relies on one specific image segmentation method, i.e., U-Net [44].

Recently, the EAP tool MicroMator [16] has emerged, strongly aligning with our goal. Similarly to the idea of smart futuristic microscopy depicted in [4] and [41], MicroMator supports reactive microscopy experiments. The developed open-source package is modular, extendable and adjustable for several experiments. However, they offer limited image analysis possibilities and no tool to improve the image analysis results. Moreover, the software seems not to be actively used and maintained.

In summary, while many tools exist that each contribute to a step towards fully EAP, no tool, to the best of our knowledge, provides a complete, modular and extendable pipeline that manages event-based data acquisition, analysis and reporting.

Segmentation. Deep learning-based segmentation methods have recently emerged as they are often faster, more accurate, and precise than traditional methods [14], given sufficient training data availability [12].

A method with pixel-wise segmentation is required to obtain all the information needed for event detection in the context of microbes. To allow the extracted data to flow directly in the EAP during the experiments, this method must be fast enough, accurate and precise to enable real-time processing [31]. Therefore, objects can be segmented, for example, with generalized methods like the SOTA vision model Segment Anything [24], so-called foundation models, which attempt to recognize all objects correctly, e.g. in the context of segmentation. Although these methods can recognize many diverse objects, they may have limitations in precision for a single use-case [22]. To overcome this problem, there are also SOTA domain-specific biomedical methods like CPN [58] and StarDist [50] or task-specific-models like Omnipose [9].

With the wide variety of models available, selecting the most appropriate one for a given task remains a significant challenge. Currently, this selection is still performed manually. However, solutions that aim to automate this selection process are being proposed. [19,35,55] investigate image similarity metrics to select the best model for a given task.

Experiment Database. MLCI experiments produce vast amounts of data. This data and associated metadata must be stored and managed for subsequent analysis and reporting. In the context of EAP4EMSIG, the data management tool must support the FAIR data management principles as depicted in [59].

For our work, the most suitable tool is Open Microscopy Environment Remote Objects (OMERO) [1], an open-source tool for managing, analyzing and visualizing microscopy images and their metadata. It offers a centralized, secure and scalable solution for handling diverse imaging data types, facilitating collaboration and data sharing among entities. Compared to other SOTA data management tools, OMERO provides advanced visualization tools and supports integration with other image analysis software [49].

Semi-Automated Data Annotations. To train the segmentation methods, corresponding training data are crucial. Annotating this data is typically time-consuming, so semi-automated segmentation tools like KaIDA [48] or ObiWan-Microbi [53] are often employed in biomedical use cases [24,47,60]. This process involves training a network with a small amount of manually annotated datasets and manually refining the network's predictions by a human annotator on new unannotated datasets.

AI-ready Image with Ground Truth Cell Simulation. A significant challenge in applying Deep Learning (DL) techniques is the need for labeled data for training and validation. Particularly in cell instance segmentation tasks, pixel-exact masks that accurately distinguish individual cells from the background are essential. Due to the high costs needed to generate such labeled data [21], cell simulators have been developed [26,56]. Their aim is to create realistic, controlled and reproducible cellular models that accurately reflect biological processes. For bacterial microcolony ground truth generation, particularly in the context of phase contrast microscopy, the cell simulator CellSium [45] emerges as the suitable tool in this work. It is an agent-based, highly customizable and versatile simulator that produces data for different use cases.

Module Interaction. Given the complexity of integrating software, hardware and biological components in laboratory experiments, a suitable architecture is required. This architecture must be robust, understandable, modular and most importantly extendable due to the uniqueness of each laboratory experiment. For our EAP, we currently use Robot Operating System (ROS) [32], an open-source framework primarily for developing robot software.

ROS provides a modular building architecture based on the central notion of nodes. Each node represents a functional unit and can exchange messages with another node, particularly in an event-based manner. Hence, ROS is very suitable for real-time tasks in various fields, such as in [20]. Nevertheless, due to the high complexity of installing and maintaining ROS as well as its dependency bugs [15], just very few approaches use it. The closest to ours is Archemist [13], an experiment-automating system for chemistry laboratories.

An alternative to ROS which is currently being investigated for our EAP is Dataflow-Oriented Robotic Architecture (DORA)[2], a framework designed to ease and simplify the development of AI-based robotic applications. To the best of our knowledge, DORA is very new and has not been used for experiment automation tasks yet. It provides low-latency, composable and distributed dataflow capabilities. Applications are organized as directed graphs, often referred to as pipelines. Although it is much faster than ROS, it is still unstable and has a rather smaller community.

3 Methodology

To fill the noted gaps, we propose a new EAP approach, which is fully described module-by-module in this section. As shown in Fig. 1, our system consists of

---

[2] https://dora-rs.ai/

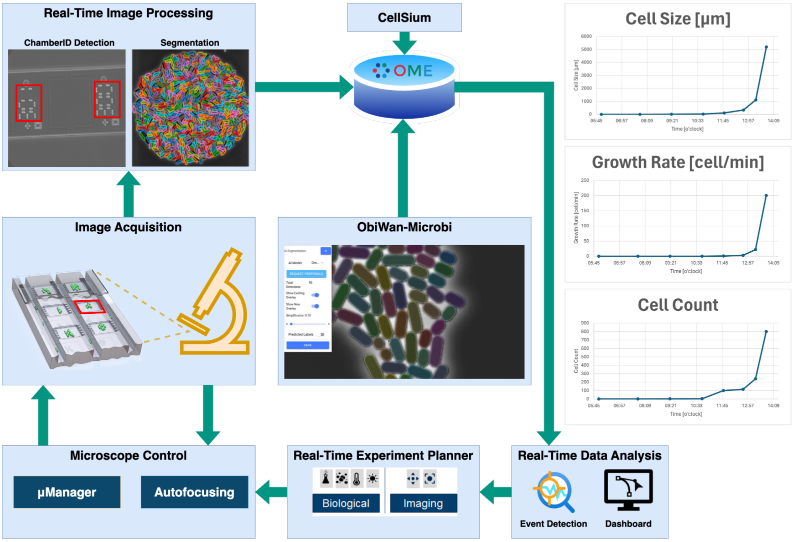


Figure 1: EAP4EMSIG visualization. The pipeline consists of eight modules, represented by the light blue boxes and the OMERO database, arranged in a cyclical process. The microbial images in the figure come from dataset [51]. The images from the experiment chip are from an internal dataset.

eight modules arranged in a cyclical process. For image acquisition, the system utilizes SOTA research microscope setups and low-cost 3D-printed microscope systems in the first EAP module. Second, the real-time image processing is executed on incoming images, generating single-cell instance segmentation predictions. The generated data and metadata are saved and managed in an instance of the third module's OMERO DataBase (DB). This instance is also used to manage ground truth data generated with the cell simulator module CellSium and the ObiWan-Microbi semi-annotation module as the fourth and fifth modules. Sixth, the real-time data analysis module relies on this data to provide feedback via a dashboard and detect events. Based on these, the real-time experiment planner, as the seventh module, schedules the experiment continuously and sends the next steps to the microscope control module, the eighth module, which forwards these instructions back to the image acquisition module. The modules are described individually in the upcoming sections.

3.1 Microscope Control

µManager. Automatic control of the microscope is essential for experiment automation. To make our EAP as independent as possible from microscope manufacturers and thus enable easy transfer to new laboratories and microscopes, µManager [11] is used. µManager is an open-source software package that controls microscopes and associated hardware components such as cameras, stages and shutters. It provides a powerful, flexible and cost-effective solution for automated microscopy. In this work, the implementation is done in Python with the help of Pymmcore(-Plus)[3] with a Nikon T1-based setup.

Autofocusing. One specific challenge we address here is the autofocusing of the microscope. We treat autofocusing as a regression problem, where a simple Multi-Layer Perceptron (MLP) is used to predict the relationship between the extracted input features from microscopy images and the continuous target variable, which is the distance to the optimal focus frame (among all z-stacks). After predicting the focus offset and direction, our system employs a closed-loop control mechanism to communicate the predicted adjustments to the microscope control. The focus is then iteratively adjusted until the optimal focus is reached.

3.2 Image Acquisition

This module handles image acquisition primarily in two different ways:

1. The most common process is using real research microscopes for the experiments and generating high-quality images. This is still by far the most used approach, particularly due to the ability of such microscopes to provide direct, high-resolution and high-fidelity images of biological samples.
2. The low-cost alternative to such expensive tools are 3D-printed microscopes, which are emerging as cost-effective and accessible tools,

---

[3] https://github.com/micro-manager/pymmcore

especially in educational settings and low-resource environments [8]. However, they generally do not match the resolution and functionality of high-end commercial microscopes.

The acquired image data and metadata are then collected and saved according to standardized protocols in a OMERO DB. Standardization offers the possibility of a uniform mask for querying different information for all modules (including future ones). Furthermore, the data can be distributed, stored and accessed worldwide.

3.3 Real-Time Image Processing

The image data acquired in the previous step (see Section 3.2) is processed as shown in Fig. 1. On the one hand, the region of interest, that is the growth chamber, is extracted by removing any microfluidic structures from the images. On the other hand, the content of the chamber is segmented using a suitable method. In this work, we focus on SOTA DL segmentation methods (see Section 2), which are either task-specific, domain-specific or foundation models and therefore allow us to address various segmentation tasks effectively. We also investigate the data processing speed of these methods. This is important because the classification of the events and, therefore, the decision of the experiment planner (see Section 3.8) is highly based on the segmentation results.

3.4 OMERO Database

As mentioned in Section 2, we use OMERO to manage not only the images and the associated metadata in a centralized and standardized manner but also the results of downstream analyses such as chamber detection and extraction, segmentation and cell analysis. All modules of the EAP (see Fig. 1) can retrieve information via a standardized interface. In addition, this makes it possible for the human expert to easily and comprehensibly document his experiments, including access to the post-processing and -analysis results.

3.5 ObiWan-Microbi

For intra- and inter-cell analysis to be possible, the best feasible extraction of objects through segmentation (see Section 3.3) is required. One challenge in our context is the large number of densely packed cells that need to be segmented. To date, there are no labeled datasets that accurately represent a comparable use case, which would facilitate transfer learning or the training of supervised segmentation methods. Since manual labeling alone would be too long and too inefficient, the semi-automated annotation tool ObiWan-Microbi is used in this work. This tool allows the prediction and correction of labels and subsequent retraining of the used DL segmentation models. An example of a dataset created with this is [51], which will be used to evaluate the segmentation methods in Section 4.

3.6 CellSium

However, even the creation of labels using semi-automated methods such as ObiWan-Microbi (see Section 3.5) costs a lot of human time and therefore money, especially in the first iteration because the segmentation methods still provide right-angled pre-segmentations. An alternative here is to have an initial basis for the segmentation methods by using automatically generated images with associated labels, e.g., from simulations. The simulator CellSium is used in our work. CellSium enables the generation of microbe images in different growth stages and also in the density and frequency required in our context. Even if these images cannot represent the full diversity of real images, combining data augmentation methods can lead to first stable results as shown in [45], where only slight adjustments have to be made in ObiWan-Microbi.

3.7 Real-Time Data Analysis

3.7.1 Dashboard

Once the microbes have been segmented, single-cell data such as average cell size and growth rate are computed and saved. This data is visualized and, most importantly, leveraged by the human expert to navigate through the experiment. For this purpose, a customized dashboard is being developed. The dashboard provides real-time insights into ongoing experiments such as cell count, growth rate and average cell size per chamber. The dashboard integrates various functionalities to facilitate the monitoring and analysis of the experiment. Due to its modular architecture, which facilitates the seamless integration of new features and functionalities without disrupting the existing codebase, our dashboard is highly extendable and can be tailored for other use cases.

3.7.2 Event Detection

The data and metadata gathered from the real-time data analysis and image processing enable us to detect different events in hundreds of parallel experiments and resolve their temporal evolution. In our case, we have two classes of events. On the one hand, technical events that are related to the devices themselves, e.g., loss of focus or chamber defects. On the other hand, we have biological events that display the behavior of microbes, such as growth rate or cell death. The detection is performed based on rules provided by the domain expert.

3.8 Real-Time Experiment Planner

A central part of the EAP pipeline is the intelligent experiment planner. The next n chambers to be explored are determined based on the last chamber recorded, including the resulting data analysis. The determination is made according to the defined experiment objectives of the human domain expert.

4 Experiments

In this section, preliminary experiments and results of our approach, particularly for real-time image segmentation, are presented and discussed. Four segmentation algorithms are compared on an Ubuntu 22.04-based workstation with an Intel Core i9-13900 Central Processing Unit (CPU), a RTX3090

Graphics Processing Unit (GPU) and a 64 GB Random-Access Memory (RAM). This system was chosen as low-performance because the hardware components represent an affordable system for users interested in such use cases. The measured inference time can be considered realistic for a lower boundary. An improved hardware configuration can achieve an additional performance boost here. We define 100 ms as the maximal limit for the real-time inference time. This is because initial tests of the microscope control program have shown that it is perfectly sufficient for the EAP4EMSIG, including autofocusing.

4.1 Dataset, Metrics and Implementation

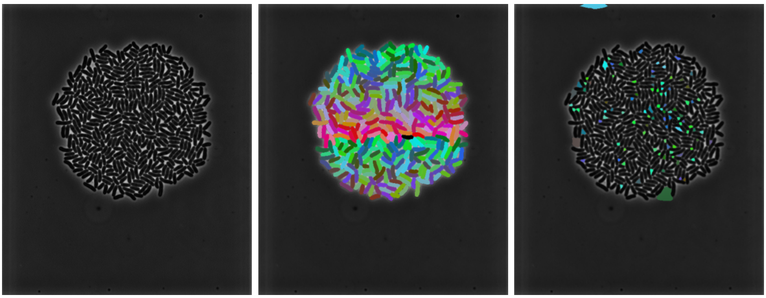
The benchmark dataset [51] is used to evaluate the methods. The dataset contains images of *Corynebacterium glutamicum* microbes and represents a typical experiment in cell culture. The dataset includes video sequences of the development of the microbes with 800 images each and consists of ground truth instance segmentation mask and tracking paths. For the context of this work, we use all 5 × 800 images purely to evaluate the segmentation performances.

To evaluate the segmentation accuracy, the metrics AP, including AP@0.50 and AP@0.75 and PQ, comprising Segmentation Quality (SQ) and Recognition Quality (RQ), are calculated for all four methods mentioned in Section 2 (see Table 1) using their respective official implementations.

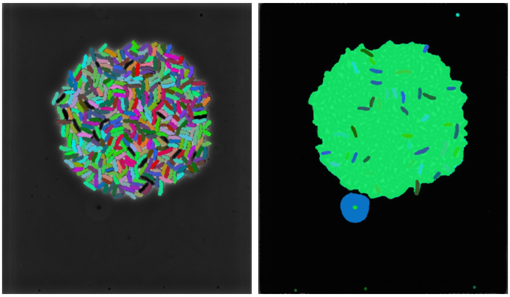
Since the AP-based metric requires the confidence score for calculation, evaluating this metric on Omnipose was impossible. Omnipose does not directly return uncertainties for predicted masks and no official instructions on how to extract these are known until the publication of the work.

4.2 Real-Time Image Processing: Segmentation

The evaluation results of the four methods are shown in Table 1. In addition, an example image from the dataset (see Fig. 2a) and the respective segmentation results (see Fig. 2b to Fig. 2e) are displayed for a medium population density with approx. 400 microbes (see Fig. 2).



(a) Original (b) Omnipose [9] (c) StarDist [50]



(d) CPN [58] (e) SAM [24]

Figure 2: Comparison of zero-shot instance segmentation predictions for [51]. The original image is shown in Fig. 2a and the predictions are shown in Fig. 2b to Fig. 2e.

From the results in Table 1, Omnipose is the best model concerning the scores PQ, SQ and RQ. However, CPN with a PQ score difference of 0.0761, i.e., domain-specific model, is not that far away and is still 86 ms faster than Omnipose. In detail, CPN has a slightly lower RQ score of 0.0526, but the difference in PQ score is primarily due to the notably worse SQ. The SQ score can be seen by directly comparing Fig. 2b and Fig. 2d. While Omnipose segmented the objects cleanly, including at the edges, CPN struggled. Additionally, some parts are often no longer properly recognized in the curved microbes towards the end.

Nevertheless, CPN's performance is quite remarkable because in CPN's training datasets, there were no such long, rod-shaped objects or similar microbial colonies in contrast to the task domain model Omnipose. As a domain model, it is also remarkable that CPN recognizes the object instances with a similarly

Method / Metric	Omnipose [9]	StarDist [50]	CPN [58]	SAM-H [24]
AP↑	-	0.0000	**0.6232**	0.0347
AP@0.5↑	-	0.0000	**0.9551**	0.0476
AP@0.75↑	-	0.0000	**0.8170**	0.0470
PQ↑	**0.9336**	0.3629	0.8575	0.0626
PQ-SQ↑	**0.9395**	0.7287	0.8779	0.8416
PQ-RQ↑	**0.9935**	0.4093	0.9763	0.0736
∅Inf. [ms]↓	271	7686	**185**	1994

Table 1: Average Precision (AP) results, Panoptic Quality (PQ) results comprising Segmentation Quality (SQ) score and Recognition Quality (RQ) score as well as inference times (Inf.) evaluated on the benchmark dataset [51].[4]

good RQ score (only 0.0172 difference). Both the number of objects and the difficult boundaries between the objects are not new for CPN and also occur, for example, in *NeurIPS 22 Cell Segmentation Competition*[5] dataset as one of the pre-training datasets. However, in combination with the microbes, this is a noteworthy generalization achievement.

The second domain model StarDist, on the other hand, with a PQ score of 0.3629 and an AP score of 0, has not yielded sufficient results and is clearly worse than CPN. The vision foundation model Segment Anything, the fourth model, is also not convincing with a PQ score of 0.0626. However, it is worth noting that the SQ score is only slightly lower than that of CPN, indicating that the objects recognized as True Positive (TP) were segmented well. However, upon examining RQ, it appears that the number of TP is likely very low. Segment Anything's problem can also be seen in Fig. 2e. There, almost the entire cluster of microbes is predicted as one object. Although there are many objects in dense clusters in the dataset SA-1B [24] on which the model was trained and also with a comparable number, it is quite possible that this transferred knowledge cannot be applied to the shape, which in turn raises doubts about the "Any" segmentation.

---

[5] https://neurips22-cellseg.grand-challenge.org/

[4] When calculating the metric, falsely detected backgrounds were not removed and evaluated during the AP calculation as false positives. The models were used according to the basic configurations for fair comparison. The values in bold are the best across all methods, provided there were results. To ensure a fair comparison, we define inference time as the duration from inputting

5 Conclusion

This paper presents the EAP4EMSIG - a novel pipeline for experiment automation for thousands of microbe colonies on microfluidic chips. For this purpose, the methodological concept of each of the eight pipeline modules was introduced, explained and distinguished from existing alternatives. Preliminary development steps of the pipeline were presented, particularly for the real-time image segmentation module. To this end, four SOTA methods were compared and evaluated quantitatively and qualitatively in the paper. CPN and Omnipose proved to be particularly powerful. Omnipose, trained task-specifically for bacteria segmentation, is 86 ms slower at inference than CPN but has a slightly better recognition quality and a noticeably higher segmentation quality. However, because CPN was not trained explicitly for bacterial segmentation, but on very diverse biomedical cells such as blood cells or nuclei of different cell types, among others, future work would investigate retraining different methods to get the best model for real-time segmentation in the EAP4EMSIG. Future work will also investigate increasing segmentation speed to achieve the minimum 100 ms required for our task, such as converting models to special inference formats like TensorRT[5] or transforming the trained models to a lower precision (e.g. FP16, Int8).

Even though only initial results for the real-time image processing module were shown in this work, the other modules are also being developed. So far, the pipeline as a whole has not yet been tested, but the modules themselves are already in advanced development and, in some cases, ready for use, such as CellSium or ObiWan-Microbi. The next steps are to combine the modules and test the EAP4EMSIG as a whole.

---

the image to receiving the model's prediction as an instance mask with confidence scores. This includes any post-processing needed by certain methods, such as converting predicted contours to a pixel-wise mask for the EAP4EMSIG pipeline. The inference time is measured with all models using FP32 precision.

[5] https://github.com/NVIDIA/TensorRT

Acknowledgments

This work was supported by the President's Initiative and Networking Funds of the Helmholtz Association of German Research Centres [Grant EMSIG ZT-I-PF-04-44]. The Helmholtz Association funds this project under the "Helmholtz Imaging Platform", the authors N. Friederich, A. J. Yamachui Sitcheu and R. Mikut under the program "Natural, Artificial and Cognitive Information Processing (NACIP)", the authors N. Friederich and A. J. Yamachui Sitcheu through the graduate school "Helmholtz Information & Data Science School for Health (HIDSS4Health)" and the author Johannes Seiffarth through the graduate school "Helmholtz School for Data Science in Life, Earth and Energy (HDS-LEE)".

The authors have accepted responsibility for the entire content of this manuscript and approved its submission. We describe here the individual contributions of N. Friederich (NF), A. J. Yamachui Sitcheu (AJYS), A. Nassal (AN), M. Pesch (MP), E. Yildiz (EY), M. Beichter (MB), L. Scholtes (LS), B. Akbaba (BA), T. Lautenschlager (TL), O. Neumann (ON), D. Kohlheyer (DK), H. Scharr (HS), J. Seiffarth (JS), K. Nöh (KN), R. Mikut (RM): Conceptualization: NF, AJYS, JS, RM; Methodology: NF, AJYS, EY, JS, DK, HS, KN, RM; Software: NF, AN, MB; Investigation: NF, AJYS, JS; Resources: JS, KN; Writing – Original Draft: NF, AJYS, MP, MB, ON, EY, JS; Writing – Review & Editing: NF, AJYS, AN, MP, EY, MB, LS, BA, TL, ON, NK, HS, JS, KN, RM; Supervision: DK, HS, KN, RM; Project administration: DK, HS, KN, RM; Funding Acquisition: DK, HS, EY, KN, RM.

References

[1] C. Allan, J.-M. Burel, J. Moore, C. Blackburn, M. Linkert, S. Loynton, D. MacDonald, W. J. Moore, C. Neves, A. Patterson, et al. OMERO: flexible, model-driven data management for experimental biology. *Nature Methods*, 9(3):245–253, 2012.

[2] U. Bethe, Z. D. Pana, C. Drosten, H. Goossens, F. König, A. Marchant, G. Molenberghs, M. Posch, P. van Damme, and O. A. Cornely. Innovative approaches for vaccine trials as a key component of pandemic preparedness - a white paper. *Infection*, 2024.

[3] K. M. Carbone, R. B. Luftig, and M. Buckley. *Microbial Triggers of Chronic Human Illness*. Am Soc Microbiol, 2005.

[4] A. E. Carpenter, B. A. Cimini, and K. W. Eliceiri. Smart microscopes of the future. *Nature Methods*, 20(7):962–964, 2023.

[5] A. Casadevall. Microbes and Climate Change - Science, People & Impacts. 2022.

[6] L. Chiron, M. Le Bec, C. Cordier, S. Pouzet, D. Milunov, A. Banderas, J.-M. Di Meglio, B. Sorre, and P. Hersen. CyberSco. Py an open-source software for event-based, conditional microscopy. *Scientific Reports*, 12(1):11579, 2022.

[7] A. Christovich and X. M. Luo. Gut microbiota, leaky gut, and autoimmune diseases. *Frontiers in Immunology*, 13:946248, 2022.

[8] J. T. Collins, J. Knapper, J. Stirling, J. Mduda, C. Mkindi, V. Mayagaya, G. A. Mwakajinga, P. T. Nyakyi, V. L. Sanga, D. Carbery, L. White, S. Dale, Z. J. Lim, J. J. Baumberg, P. Cicuta, S. McDermott, B. Vodenicharski, and R. Bowman. Robotic microscopy for everyone: the OpenFlexure microscope. *Biomed. Opt. Express*, 11(5):2447–2460, 2020.

[9] K. J. Cutler, C. Stringer, T. W. Lo, L. Rappez, N. Stroustrup, S. Brook Peterson, P. A. Wiggins, and J. D. Mougous. Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation. *Nature Methods*, 19(11):1438–1448, 2022.

[10] P. Dettinger, T. Frank, M. Etzrodt, N. Ahmed, A. Reimann, C. Trenzinger, D. Loeffler, K. D. Kokkaliaris, T. Schroeder, and S. Tay. Automated microfluidic system for dynamic stimulation and tracking of single cells. *Analytical Chemistry*, 90(18):10695–10700, 2018.

[11] A. D. Edelstein, M. A. Tsuchida, N. Amodaj, H. Pinkard, R. D. Vale, and N. Stuurman. Advanced methods of microscope control using μManager software. *Journal of Biological Methods*, 1(2), 2014.

[12] A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean, and R. Socher. Deep learning-enabled medical computer vision. *npj Digital Medicine*, 4(1):5, 2021.

[13] H. Fakhruldeen, G. Pizzuto, J. Glowacki, and A. I. Cooper. Archemist: Autonomous robotic chemistry system architecture. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6013–6019. IEEE, 2022.

[14] M. S. Fasihi and W. B. Mikhael. Overview of current biomedical image segmentation methods. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 803–808, 2016.

[15] A. Fischer-Nielsen, Z. Fu, T. Su, and A. Wąsowski. The forgotten case of the dependency bugs: on the example of the robot operating system. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice*, pages 21–30, 2020.

[16] Z. R. Fox, S. Fletcher, A. Fraisse, C. Aditya, S. Sosa-Carrillo, J. Petit, S. Gilles, F. Bertaux, J. Ruess, and G. Batt. Enabling reactive microscopy with MicroMator. *Nature Communications*, 13(1):2199, 2022.

[17] N. Friederich, A. J. Yamachui Sitcheu, O. Neumann, S. Eroglu-Kayıkçı, R. Prizak, L. Hilbert, and R. Mikut. AI-based automated active learning for discovery of hidden dynamic processes: A use case in light microscopy. In *Proceedings-33. Workshop Computational Intelligence: Berlin, 23.-24. November 2023*, volume 23, page 31. KIT Scientific Publishing, 2023.

[18] M. Ganesan, R. Mani, S. Sai, G. Kasivelu, M. K. Awasthi, R. Rajagopal, N. I. Wan Azelee, P. K. Selvi, S. W. Chang, and B. Ravindran. Bioremediation by oil degrading marine bacteria: An overview of

supplements and pathways in key processes. *Chemosphere*, 303(Pt 1):134956, 2022.

[19] P. Godau and L. Maier-Hein. Task Fingerprinting for Meta Learning in Biomedical Image Analysis. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2021*, pages 436–446. Springer, 2021.

[20] F. He and L. Zhang. Design of indoor security robot based on robot operating system. *Journal of Computer and Communications*, 11(5):93–107, 2023.

[21] H. Jeckel and K. Drescher. Advances and opportunities in image analysis of bacterial cells and communities. *FEMS Microbiology Reviews*, 45, 2020.

[22] W. Ji, J. Li, Q. Bi, T. Liu, W. Li, and L. Cheng. Segment Anything Is Not Always Perfect: An Investigation of SAM on Different Real-world Applications. *Machine Intelligence Research*, 21(4):617–630, Aug 2024.

[23] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[24] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment Anything. *arXiv:2304.02643*, 2023.

[25] G. Lancini and A. L. Demain. Bacterial pharmaceutical products. In E. Rosenberg, E. F. DeLong, S. Lory, E. Stackebrandt, and F. Thompson, editors, *The Prokaryotes*, pages 257–280. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[26] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, and O. Yli-Harja. Computational framework for simulating fluorescence microscope images with cell populations. *IEEE Transactions on Medical Imaging*, 26(7):1010–1016, 2007.

[27] M. Li, A. Fang, X. Yu, K. Zhang, Z. He, C. Wang, Y. Peng, F. Xiao, T. Yang, W. Zhang, X. Zheng, Q. Zhong, X. Liu, and Q. Yan. Microbially-driven sulfur cycling microbial communities in different mangrove sediments. *Chemosphere*, 273:128597, 2021.

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[29] B.-N. Liu, X.-T. Liu, Z.-H. Liang, and J.-H. Wang. Gut microbiota in obesity. *World Journal of Gastroenterology*, 27(25):3837–3850, 2021.

[30] K. J. Locey and J. T. Lennon. Scaling laws predict global microbial diversity. *Proceedings of the National Academy of Sciences of the United States of America*, 113(21):5970–5975, 2016.

[31] A. Lou, S. Guan, and M. Loew. CFPNet-M: A Light-Weight Encoder-Decoder Based Network for Multimodal Biomedical Image Real-Time Segmentation. *Computers in Biology and Medicine*, 154:106579, 2023.

[32] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall. Robot Operating System 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66):eabm6074, 2022.

[33] D. Mahecic, W. L. Stepp, C. Zhang, J. Griffié, M. Weigert, and S. Manley. Event-driven acquisition for content-enriched microscopy. *Nature Methods*, 19(10):1262–1267, 2022.

[34] B. Miles and P. L. Lee. Achieving Reproducibility and Closed-Loop Automation in Biological Experimentation with an IoT-Enabled Lab of the Future. *SLAS TECHNOLOGY: Translating Life Sciences Innovation*, 23(5):432–439, 2018.

[35] M. Molina-Moreno, M. P. Schilling, M. Reischl, and R. Mikut. Automated Style-Aware Selection of Annotated Pre-Training Databases in Biomedical Imaging. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5, 2023.

[36] J. M. Nduko and S. Taguchi. Microbial production of biodegradable lactate-based polymers and oligomeric building blocks from renewable and waste resources. *Frontiers in Bioengineering and Biotechnology*, 8:618077, 2020.

[37] J. P. Neto, A. Mota, G. Lopes, B. J. Coelho, J. Frazão, A. T. Moura, B. Oliveira, B. Sieira, J. Fernandes, E. Fortunato, R. Martins, R. Igreja, P. V. Baptista, and H. Águas. Open-source tool for real-time and automated analysis of droplet-based microfluidic. *Lab Chip*, 23:3238–3244, 2023.

[38] K. Oliphant and E. Allen-Vercoe. Macronutrient metabolism by the human gut microbiome: major fermentation by-products and their impact on host health. *Microbiome*, 7(1):91, 2019.

[39] E. Pedone, I. De Cesare, C. G. Zamora-Chimal, D. Haener, L. Postiglione, A. La Regina, B. Shannon, N. J. Savery, C. S. Grierson, M. Di Bernardo, et al. Cheetah: a computational toolkit for cybergenetic control. *ACS Synthetic Biology*, 10(5):979–989, 2021.

[40] H. Pinkard, N. Stuurman, I. E. Ivanov, N. M. Anthony, W. Ouyang, B. Li, B. Yang, M. A. Tsuchida, B. Chhun, G. Zhang, et al. Pycro-Manager: open-source software for customized and reproducible microscope control. *Nature Methods*, 18(3):226–228, 2021.

[41] H. Pinkard and L. Waller. Microscopes are coming for your job. *Nature Methods*, 19(10):1175–1176, 2022.

[42] D. M. S. Pinto, M. A. Phillips, N. J. Hall, J. Mateos-Langerak, D. Stoychev, T. S. Pinto, M. J. Booth, I. Davis, and I. M. Dobbie. Python-Microscope – a new open-source Python library for the control of microscopes. *Journal of Cell Science*, 134, 2021.

[43] F. Rahmanian, J. Flowers, D. Guevarra, M. Richter, M. Fichtner, P. Donnely, J. M. Gregoire, and H. S. Stein. Enabling modular autonomous feedback-loops in materials science through hierarchical experimental laboratory automation and orchestration. *Advanced Materials Interfaces*, 9(8):2101987, 2022.

[44] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer, 2015.

[45] C. C. Sachs, K. Ruzaeva, J. Seiffarth, W. Wiechert, B. Berkels, and K. Nöh. CellSium: versatile cell simulator for microcolony ground truth generation. *Bioinformatics Advances*, 2(1):vbac053, 2022.

[46] S. Sanchez and A. L. Demain. Useful microbial enzymes—an introduction. In *Biotechnology of Microbial Enzymes*, pages 1–11. Elsevier, 2017.

[47] T. Scherr, J. Seiffarth, B. Wollenhaupt, O. Neumann, M. P. Schilling, D. Kohlheyer, H. Scharr, K. Nöh, and R. Mikut. microbeSEG: A deep learning software tool with OMERO data management for efficient and accurate cell segmentation. *Plos one*, 17(11):e0277601, 2022.

[48] M. P. Schilling, S. Schmelzer, L. Klinger, and M. Reischl. KaIDA: a modular tool for assisting image annotation in deep learning. *Journal of Integrative Bioinformatics*, 19(4):20220018, 2022.

[49] C. Schmidt, J. Hanne, J. Moore, C. Meesters, E. Ferrando-May, S. Weidtkamp-Peters, et al. Research data management for bioimaging: the 2021 NFDI4BIOIMAGE community survey. *F1000Research*, 11, 2022.

[50] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers. Cell Detection with Star-Convex Polygons. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2018*, pages 265–273. Springer, 2018.

[51] J. Seiffarth, L. Blöbaum, K. Löffler, T. Scherr, A. Grünberger, H. Scharr, R. Mikut, and K. Nöh. Data for - Tracking one in a million: Performance of automated tracking on a large-scale microbial data set. `https://doi.org/10.5281/zenodo.7260137`, 10 2022.

[52] J. Seiffarth, L. Blöbaum, R. Paul, N. Friederich, A. J. Yamachui Sitcheu, R. Mikut, H. Scharr, A. Grünberger, and K. Nöh. Tracking one-in-a-million: Large-scale benchmark for microbial single-cell tracking

with experiment-aware robustness metrics. In *European Conference on Computer Vision*. Springer, 2024.

[53] J. Seiffarth, T. Scherr, B. Wollenhaupt, O. Neumann, H. Scharr, D. Kohlheyer, R. Mikut, and K. Nöh. ObiWan-Microbi: OMERO-based integrated workflow for annotating microbes in the cloud. *SoftwareX*, 26:101638, 2024.

[54] R. Sender, S. Fuchs, and R. Milo. Revised estimates for the number of human and bacteria cells in the body. *PLoS Biology*, 14(8):e1002533, 2016.

[55] A. Y. Sitcheu, N. Friederich, S. Baeuerle, O. Neumann, M. Reischl, and R. Mikut. MLOps for Scarce Image Data: A Use Case in Microscopic Image Analysis. In *Proceedings-33. Workshop Computational Intelligence: Berlin, 23.-24. November 2023*, volume 23, page 169. KIT Scientific Publishing, 2023.

[56] D. Svoboda and V. Ulman. MitoGen: A Framework for Generating 3D Synthetic Time-Lapse Sequences of Cell Populations in Fluorescence Microscopy. *IEEE Transactions on Medical Imaging*, 36:310–321, 2017.

[57] D. M. Sylvia, J. J. Fuhrmann, P. G. Hartel, and D. A. Zuberer. *Principles and applications of soil microbiology*. Pearson, 2005.

[58] E. Upschulte, S. Harmeling, K. Amunts, and T. Dickscheid. Contour proposal networks for biomedical instance segmentation. *Medical Image Analysis*, 77:102371, 2022.

[59] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1):1–9, 2016.

[60] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, and L. Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829, 2024.

Optimal Scaling of an Algorithmic Parameter in Restart Strategies

Lisa Schönenberger, Hans-Georg Beyer

Vorarlberg University of Applied Sciences
Research Center Business Informatics
6850 Dornbirn, Austria
E-Mail: {lisa.schoenenberger, hans-georg.beyer}@fhv.at

Abstract

This paper investigates restart strategies for algorithms whose success depends on an algorithmic parameter λ. It is assumed that there exists a unique unknown optimal λ. After each restart λ is increased. The main question is whether there is an optimal strategy for choosing λ after each restart. To this end, possible restart strategies are classified into parameter-dependent strategy types. A loss function is introduced, that measures the wasted computational costs compared to the optimal strategy. One criterion that a viable restart strategy must satisfy is that the loss relative to the optimal λ is bounded. Experimental evidence demonstrates that this is not the case for all strategy types. However, for a specific strategy type, where the parameter λ is increased multiplicatively with an increasing constant ρ, the relative loss function has an upper bound. It will be shown, that for this strategy type there is an optimal choice for the parameter ρ that is independent of the optimal λ.

This work was supported by the Austrian Science Fund (FWF) under grant P33702-N.

1 Introduction

Restart strategies are techniques used to restart an algorithm after it has failed. Beyond simple random restarts, there are several approaches to implement such strategies. One common method is to restart the algorithm from a different start point, as it is usual for gradient strategies in nonlinear optimization. If the performance of the algorithm depends on a certain parameter, this parameter can be adjusted after each restart. It quickly becomes clear that the exact implementation of a restart strategy is highly problem-specific, therefore restrictions must be made when analyzing restart strategies. This paper focuses on algorithms whose success depends on a single parameter λ that is modified after each restart according to a predetermined rule. The primary goal of this study is to evaluate these rules, called strategy types, in order to identify an optimal restart strategy. Optimality criteria have already been studied for algorithms, whose success depends on the execution time, see for example [1] or [2]. For the restart strategies of parameter-dependent algorithms, optimality has not been studied so far.

The specific constraints relevant to the algorithms studied are outlined in Section 2. After a general definition of a restart strategy in Section 3, the various strategy types are introduced. Section 4 introduces the concept of loss and presents the relative loss function, with the results of related experiments. The paper concludes with a summary of the results and an outlook.

2 Preliminary conditions

The restart strategy (RS) considered here are applied to algorithms that satisfy the following constraints. The algorithm **A** depends on an algorithmic parameter $\lambda \in \mathbb{N}$. **A** is successful only if this algorithmic parameter exceeds a certain bound $\hat{\lambda}$. Formally, this means

$$\begin{aligned} &\mathbf{A}(\lambda) \quad \text{is successful if} \quad \lambda \geq \hat{\lambda} \\ &\mathbf{A}(\lambda) \quad \text{is unsuccessful if} \quad \lambda < \hat{\lambda}. \end{aligned} \tag{1}$$

$F_E(\lambda)$ measures the computational costs. Since evolution strategies are considered it is defined by the number of objective function evaluations that $\mathbf{A}(\lambda)$ uses until termination. It is assumed that $F_E(\lambda)$ increases with λ, so that the optimal choice is to execute the algorithm with $\hat{\lambda}$. Therefore, $\hat{\lambda}$ is also called *optimal* λ. An additional assumption is that

$$F_E(\lambda) = g\lambda \tag{2}$$

for a constant $g \in \mathbb{N}$.

Algorithms that satisfy these conditions roughly are for example Evolution Strategies in multimodal landscapes. The parameter λ in this case is the population size. There is an interval for λ where a positive success probability less than 1 is possible. This interval, however, is small compared to the population size. If λ exceeds this interval, the success rate remains constant at one. This has been demonstrated for several multimodal test functions in [3]. The validity of the simplification $F_E(\lambda) = g$ was shown in [4].

3 Restart Strategies

To approach the optimal choice of the algorithmic parameter $\hat{\lambda}$, which is generally unknown, a restart strategy can be used. Restart strategies (RS) are defined by an infinite sequence

$$\mathbb{R} = (\lambda_0, \lambda_1, \lambda_2, \ldots), \quad \lambda_k \in \mathbb{N}, \tag{3}$$

where λ_k is the algorithmic parameter of the kth run. The kth run of the RS is denoted by $\mathbf{R}_k$. $\mathbf{R}_k$ is stopped when a local stopping criterion is satisfied. Then, an independent algorithm $\mathbf{R}_{k+1}$ with parameter λ_{k+1} is executed. This process is repeated until the algorithm is successful.

The theory of restart strategies raises the question of how to choose λ_k. Because of condition (1), it is clear that λ should be increased after each restart. In principle, there are an infinite number of restart strategies. A common choice for λ_k is $\lambda_k = \lambda_0 2^k$ (see for example [5] or [6]). In this case, λ is increased multiplicatively after each restart. This type of a restart strategy will be also examined in the following. Instead of base 2, other increasing factors may be used, which leads to the following restart sequence

$$\begin{aligned} \mathbb{R}^{\times} &= (\lambda_0, \lambda_1, \lambda_2, \ldots) \\ \lambda_k &= \lceil \lambda_0 \rho^k \rceil. \end{aligned} \tag{4}$$

$\mathbb{R}^{\times}$ is called a *strategy type* and ρ is called *increasing constant*. Because of the assumption that λ increases after each restart, it can be assumed that $\rho > 1$. ρ does not necessarily have to be a natural number, therefore, the amount is rounded up.

For $\mathbb{R}^{\times}$-RS, the rounding of λ_k occurs only once at the end. Alternatively, one can consider the $\mathbb{R}^{*}$-RS where λ_k is determined based on the previous rounded-up values, resulting in

$$\begin{aligned} \mathbb{R}^{*} &= (\lambda_0, \lambda_1, \lambda_2, \ldots) \\ \lambda_k &= \lceil \lambda_{k-1} \rho \rceil = \lceil \lceil \lceil \lambda_0 \rho \rceil \rho \rceil \ldots \rho \rceil, \quad k \geq 1, \end{aligned} \tag{5}$$

$\mathbb{R}^{\times}$ and $\mathbb{R}^{*}$ are multiplicative strategy types. Another type increases the population size by a constant amount, i.e.,

$$\begin{aligned} \mathbb{R}^{+} &= (\lambda_0, \lambda_1, \lambda_2, \ldots) \\ \lambda_k &= \lambda_{k-1} + \nu = \lambda_0 + k\nu, \quad k \geq 1, \nu \in \mathbb{N} \setminus \{0\}. \end{aligned} \tag{6}$$

A third type of restart strategies obeys a power law with constant $\alpha \geq 1$ defined by

$$\begin{aligned} \mathbb{R}^{\#} &= (\lambda_0, \lambda_1, \lambda_2, \ldots) \\ \lambda_k &= \lceil \lambda_0 (k+1)^{\alpha} \rceil. \end{aligned} \tag{7}$$

To illustrate the impact of the different restart strategies, exemplary runs of an Evolution Strategy with cumulative stepsize adaptation (CSA-ES) [7] on the well-known Rastrigin function are presented in Fig. 1. The exact definition of the CSA-ES and the Rastrigin function is shown in the Appendix. The figure

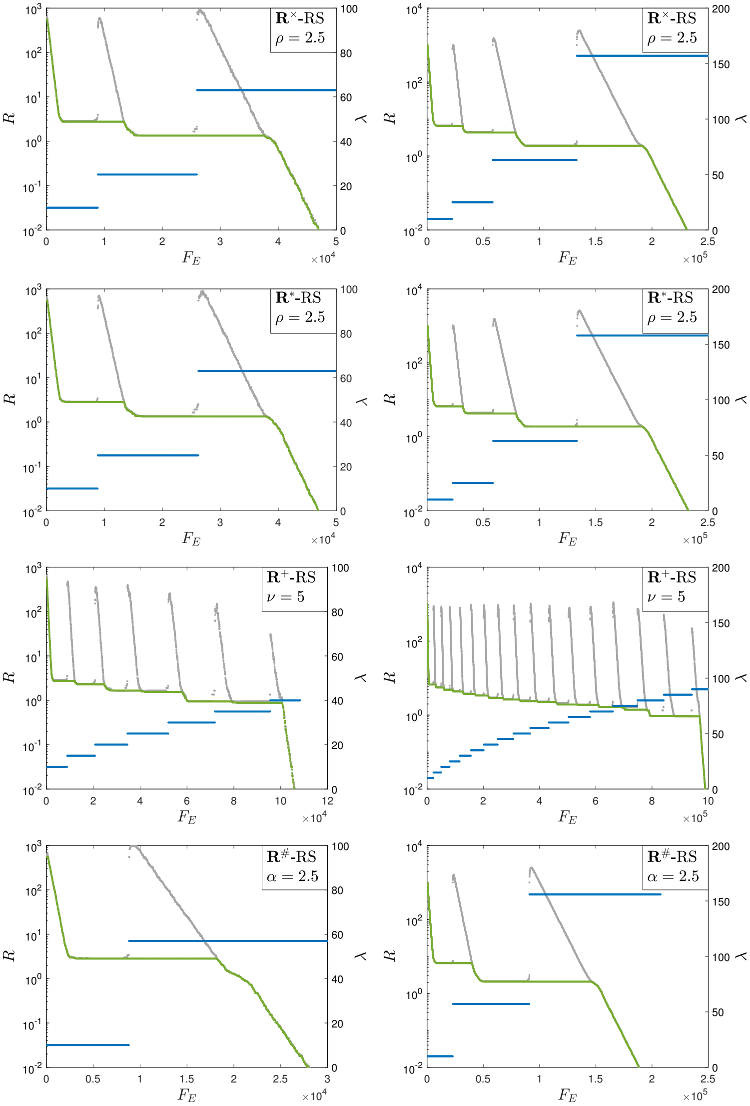


Figure 1: R-dynamics of the $\mathbb{R}$-RS (3) applied to the CSA-ES (Alg. 2) in the Rastrigin landscape (18) with $\vartheta = 0.5$, $N = 30$ (left plots) and $N = 100$ (right plots). All experiments use $\lambda_0 = 10$ as start value. R denotes the distance to the global optimizer. F_E denotes the number of function evaluations. Gray markers show the median of R, green markers show the median of the best R-value achieved up to F_E. Blue markers show the median of λ. The median was taken over 51 independent runs.

depicts the R-dynamics of restart strategies as a function of the number of function evaluations. The total number of function evaluations was calculated by aggregating the data across all restarts. The variable R represents the distance to the global optimizer and is depicted by the gray markers. The green markers indicate the best value of R that has been attained so far. Both are represented by the left y-axis. The blue markers show the population size λ, which is represented by the right y-axis. The λ-values represent a step function. At the jumps of this step function, a new restart begins with a larger value of λ. In the associated R-dynamics, one observes that a smaller distance to the optimum can be achieved after a certain number of function evaluations following the restart. This results in the emergence of kinks in the R-dynamics. The left plot employs $N = 30$, while the right plots use $N = 100$. For larger dimensions, a larger value of $\hat{\lambda}$ is necessary to achieve success. This is visible in Fig. 1 where the number of function evaluations and the number of restarts are larger in the right figure.

A comparative analysis of the various strategy types reveals that the $\mathbb{R}^{+}$-RS requires significantly more restarts in comparison to the other types. Moreover, the number of function evaluations required to achieve success is considerably larger. However, this could be compensated for by choosing a larger value for ν. The multiplicative strategy types $\mathbb{R}^{*}$ and $\mathbb{R}^{\times}$ exhibit no significant differences. However, it should be noted that the value of λ is larger for the $\mathbb{R}^{*}$-RS due to the additional rounding. This effect becomes more pronounced the more restarts are required. Interestingly, the strategy type $\mathbb{R}^{\#}$ clearly has the best performance in this example.

It is not obvious whether one strategy type is better than the other. It is also not clear how to choose the increasing constants ν, ρ, and α for any given strategy type. There is no optimality criterion up until now. In the following sections, the influence of the increasing constants on the number of function evaluations is investigated experimentally.

4 The Loss Function

When choosing the increasing constant, it is important to avoid choosing values that are too small as this will result in many restarts being necessary. Conversely, if the increasing constant is set to a very large value, λ will also become very large after just a few restarts. This causes λ to be much larger than necessary, requiring more function evaluations than necessary. The loss ΔF_E of a restart strategy is defined by[1]

$$\Delta F_E\left(\hat{\lambda},\rho\right) := \sum_{k=0}^{\hat{k}(\hat{\lambda})} F_E(\lambda_k) - F_E(\hat{\lambda}) = \left(\sum_{k=0}^{\hat{k}(\hat{\lambda})} \lambda_k - \hat{\lambda}\right) g, \tag{8}$$

where the second equality follows from condition (2). $\hat{k}(\hat{\lambda})$ denotes the minimum number of restarts required to obtain a λ larger than or equal to $\hat{\lambda}$, i.e.,

$$\hat{k}\left(\hat{\lambda}\right) := \arg\min\{k|\lambda_k \geq \hat{\lambda}\}. \tag{9}$$

Because g is constant it can be dropped in the following considerations. Therefore, the reduced loss function

$$\mathbf{L}\left(\hat{\lambda},\rho\right) := \sum_{k=0}^{\hat{k}(\hat{\lambda})} \lambda_k - \hat{\lambda}. \tag{10}$$

will be used in the following. (10) can be calculated numerically using Alg. 1. The update of λ depends on the specific strategy type.

Algorithm 1.: Numerical Calculation of the Loss Function (10)

```
Initialize ($\lambda = \lambda_0, F_E = \lambda_0, k = 0$)
while $\lambda < \hat{\lambda}$ do
    $k = k + 1$
    $\lambda = r(\lambda)$                    ▷ update $\lambda$, depending on strategy type
    $F_E = F_E + \lambda$
$\mathbf{L} = F_E - \hat{\lambda}$
```

---

[1] ρ is used as a substitute to indicate the dependency of ΔF_E on the respective increasing constants.

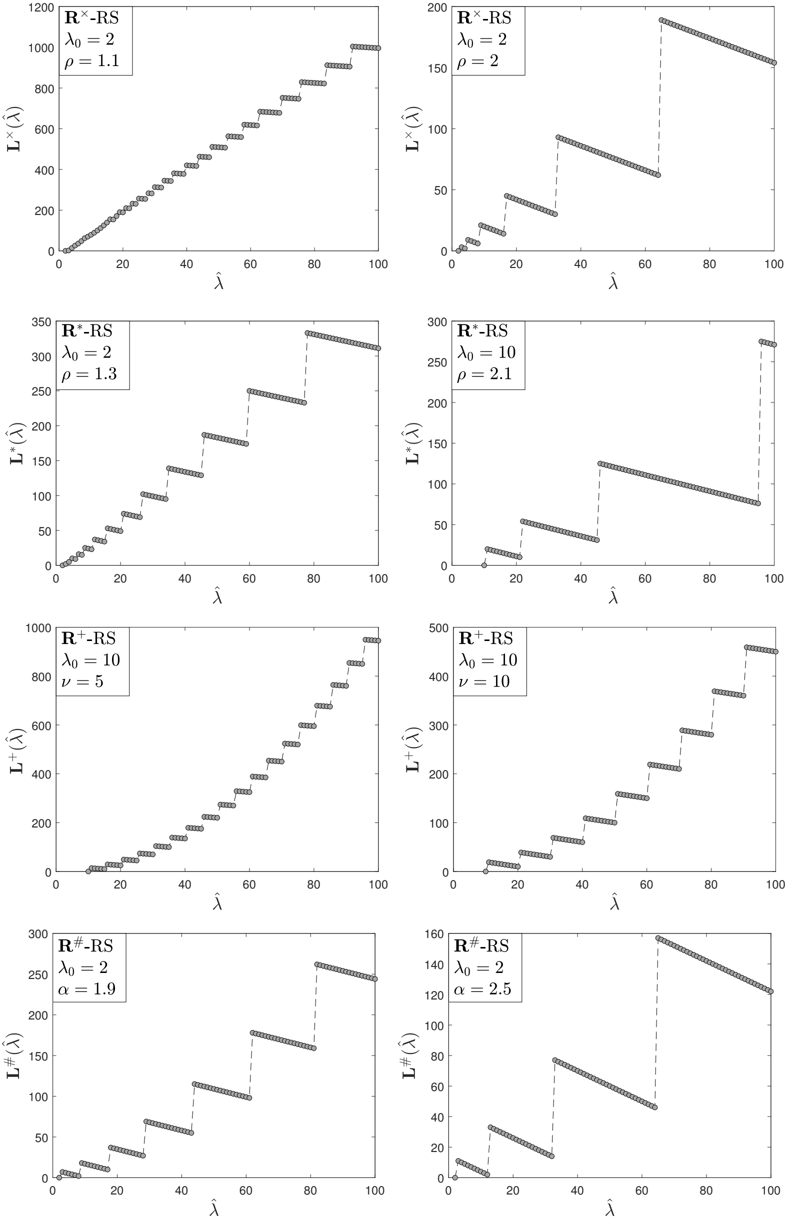


Figure 2: Loss function (10) determined numerically using Alg. 1.

Figure 2 shows loss functions as a function of $\hat{\lambda}$ for a fixed value of the increasing constant. The markers represent the numerical solution for the loss function computed with Alg. 1 and

$$
\begin{aligned}
r(\lambda) &= \lceil \lambda_0 \rho^k \rceil \quad \text{for} \quad \mathbb{R}^\times && (11)\\
r(\lambda) &= \lceil \lambda \rho \rceil \quad \text{for} \quad \mathbb{R}^* && (12)\\
r(\lambda) &= \lambda + \nu \quad \text{for} \quad \mathbb{R}^+ && (13)\\
r(\lambda) &= \lceil \lambda_0 (k+1)^\alpha \rceil \quad \text{for} \quad \mathbb{R}^\#, && (14)
\end{aligned}
$$

where k is the number of restarts. For all strategy types, the loss function jumps. These jumps occur at the values $\lambda_k + 1$ where the cost of an additional restart is added. Between these jumps, the loss decreases linearly with $\hat{\lambda}$. The left and right plots show the same strategy type with different increasing constants. The frequency of the jumps is larger for smaller increasing constants. The magnitude of the loss varies considerably for the different increasing constants.

It can also be seen from Fig. 2 that for a fixed value of the increasing constant, the loss is unbounded of $\hat{\lambda}$. The larger $\hat{\lambda}$ is, the larger the loss can be expected. Therefore, to further characterize the restart effort, it is useful to introduce the relative loss. It measures the loss relative to $\hat{\lambda}$. The relative loss is defined by

$$
\ell(\hat{\lambda}, \rho) := \frac{\mathbf{L}(\hat{\lambda}, \rho)}{\hat{\lambda}}. \tag{15}
$$

The relative loss function is shown in Fig. 3 for all strategy types. This function also has the characteristic jumps and decreases in $\hat{\lambda}$ between these jumps. The top plots represent the $\mathbb{R}^+$-RS and the $\mathbb{R}^\#$-RS. For both cases and for sufficiently large $\hat{\lambda}$, the relative loss is observed to be smaller for larger increasing constants. In all cases, the relative loss goes to infinity with $\hat{\lambda}$. In contrast, for the multiplicative strategy types $\mathbb{R}^\times$ and $\mathbb{R}^*$, represented by the bottom plots, this is not the case. The relative loss function has an upper bound. The figures demonstrate that the relative loss functions for the $\mathbb{R}^\times$-RS and $\mathbb{R}^*$-RS are nearly identical, which suggests that both strategies can be considered interchangeable.

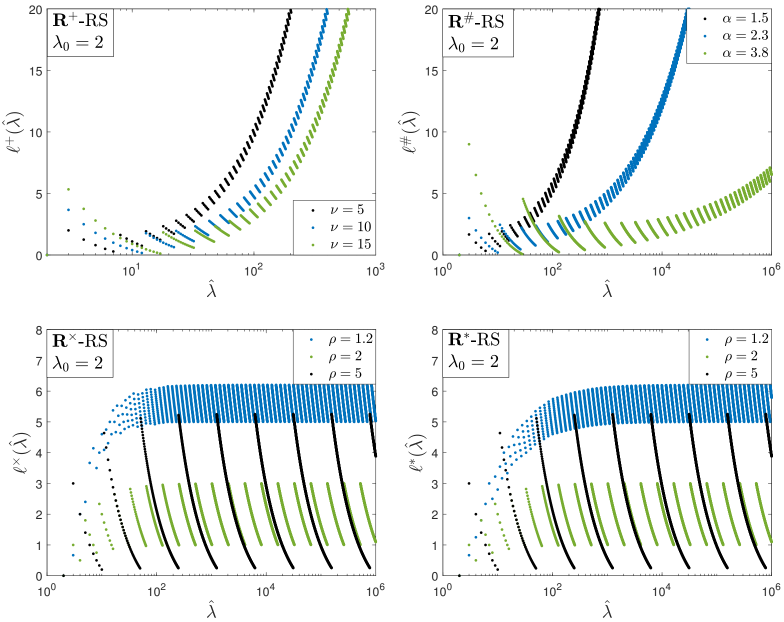


Figure 3: Relative loss function (15) determined numerically using Alg. 1.

If the relative loss has an upper bound $\ell_{\text{up}}(\hat{\lambda}, \rho)$, the strategy is classified as *bounded*. Consequently, the $\mathbb{R}^{\times}$-RS and the $\mathbb{R}^{*}$-RS are bounded restart strategies. In contrast, the $\mathbb{R}^{+}$-RS and the $\mathbb{R}^{\#}$-RS are not bounded.

The upper bound of the multiplicative restart strategies depends strongly on ρ. The local maxima of the relative loss functions approach a constant value as $\hat{\lambda}$ increases. The value

$$\overline{\ell_{\text{up}}}(\rho) := \lim_{\hat{\lambda} \to \infty} \ell_{\text{up}}(\hat{\lambda}, \rho) \tag{16}$$

is called the *asymptotic upper bound* of the relative loss function. This upper bound is dependent on the increasing constant. The objective is to find the value of ρ that minimizes the asymptotic upper bound. This value is called the optimal choice of the increasing constant and is independent of $\hat{\lambda}$. This is explored further for the $\mathbb{R}^{\times}$-RS in Fig. 4.

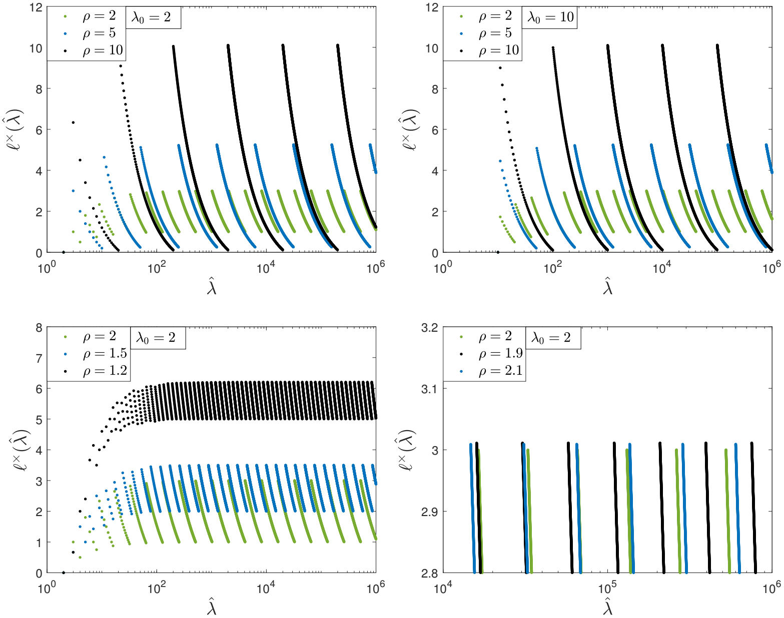


Figure 4: Relative loss function (15) for the $\mathbb{R}^{\times}$-RS determined numerically using Alg. 1.

The upper left and upper right plots illustrate the relative loss function for identical values of ρ, but for different start values λ_0. The left plot uses $\lambda_0 = 2$ and the right plot $\lambda_0 = 10$, respectively. The asymptotic upper bounds are identical in both plots, indicating that they are independent of λ_0. As illustrated in the upper plots of Fig. 4, the asymptotic upper bound of the relative loss function is larger for larger values of ρ. Conversely, the lower left plot shows that the asymptotic upper bound is smaller for larger ρ. In both cases, the upper bound attains its minimal value when $\rho = 2$. The lower right plot shows the relative loss where ρ has values closed at 2. Again, the loss is minimal when ρ is equal to 2. This leads to the hypothesis that $\hat{\rho} = 2$ is the optimal choice of ρ. The asymptotic upper bound of the relative loss function in this case is 3. Based on these observations, the following theorem can be formulated:

Theorem: The asymptotic upper bound $\overline{\ell_{\mathrm{up}}^{\times}}(\rho)$ (16) of the relative loss for the $\mathbb{R}^{\times}$-RS is minimal for $\hat{\rho} = 2$. Furthermore, it holds

$$\overline{\ell_{\mathrm{up}}^{\times}}(\hat{\rho}) = 3. \tag{17}$$

The proof will be given in a forthcoming paper.

In addition to the asymptotic upper bound, one may also define an asymptotic lower bound. As illustrated in the upper plots of Fig. 3, the lower bound of the relative loss for the $\mathbb{R}^{+}$-RS and the $\mathbb{R}^{\#}$-RS also tend to infinity. In the case of the $\mathbb{R}^{\times}$-RS, the asymptotic lower bound is observed to be smaller, the larger ρ is. This is visible in the left plots of Fig. 4 and indicates that there is no optimal choice of ρ w.r.t. the asymptotic lower bound.

5 Conclusion and Future Work

This paper investigated restart strategies (RS) applied to algorithms whose success depends on an algorithmic parameter λ. The optimal choice of this algorithmic parameter for RS is a question that has not been investigated so far. To this end, the loss function was introduced, which measures the number of function evaluations of an RS compared to the number of function evaluations of the optimal strategy. Even more important is the relative loss, i.e., the loss relative to the optimal λ. To estimate and compare the relative loss for different restart strategies, the set of all restart strategies was divided into parameter-dependent subsets called strategy types. The goal was to find strategy types whose relative loss functions are upper bounded. In this case, it is possible to minimize the upper bound according to the parameter of the strategy type.

To gain further insight, the relative loss function was determined experimentally for different strategy types. In the case of the $\mathbb{R}^{+}$-RS, where the same amount ν is added after each restart, the relative loss function is unbounded. The same is true for the $\mathbb{R}^{\#}$, where the parameters are determined according to a power law. For the multiplicative strategy type $\mathbb{R}^{\times}$-RS, where the algorithmic parameter is multiplied by an increasing constant ρ after each restart, it was demonstrated that the relative loss function is bounded. The same is true for the $\mathbb{R}^{*}$-RS, which is also a multiplicative strategy type. In contrast to the $\mathbb{R}^{\times}$-RS the value of λ_k for this strategy type is derived from the previous rounded values. In investigating this strategy type, it was shown that there exists an optimal choice of ρ, which minimizes the asymptotic upper bound of the relative loss function. The optimal choice for the increasing constant is $\hat{\rho} = 2$. When $\rho = \hat{\rho}$ is chosen, the maximum relative loss w.r.t. $\hat{\lambda}$ is 3. These results are independent of the start value λ_0. It was also demonstrated that there is no ρ that minimizes the asymptotic lower bound of the relative loss function. All of these results have been demonstrated experimentally. The analytical details and proofs will be presented in a forthcoming paper.

Appendix

CSA-Evolution Strategy: The experiments described in Section 3 are conducted using the $(\mu/\mu_I, \lambda)$-CSA-ES with $c = 1/\sqrt{N}$. This Evolution Strategy (ES) is outlined in detail in Alg. 2. In each generation, the population comprises λ offspring individuals. The offspring individuals are generated with mutation strength σ by the use of isotropic Gaussian mutations (Lines 4 and 5). In the selection step, the fitness value of each offspring individual is calculated (Line 6). Subsequently, the individuals are sorted in accordance with their fitness values (Line 7). Only the μ individuals with the best fitness survive and serve as the parents for the subsequent generation. The ratio between parents and offspring is denoted by $\vartheta := \mu/\lambda$ and is referred to as the truncation ratio. The initial state of the next generation is the result of recombination, i.e., the centroid of all parents (Line 9). The subscript $m;\lambda$ denotes the mth best individual from λ offspring. The mutation strength σ is adapted for each new generation through cumulative step-size adaptation (CSA) (Lines 10 - 12). c is called the cumulation time parameter.

Test Function: The Rastrigin function for an N-dimensional search vector $\mathbf{y} = (y_1, \ldots, y_N)$ is given by

$$F(\mathbf{y}) = \sum_{i=1}^{N} \left[y_i^2 + A\left(1 - \cos(\alpha y_i)\right) \right] \tag{18}$$

Algorithm 2.: The $(\mu/\mu_I,\lambda)$-CSA Evolution Strategy

```
Initialize $\left(\mathbf{y}^{(0)}, \sigma^{(0)}, \sigma_{\text{stop}}, \mathbf{s}=\mathbf{1}, g=0\right)$
repeat
    for $l=1$ to $\lambda$ do
        $\tilde{\mathbf{z}}_l = (\mathcal{N}(0,1),\dots,\mathcal{N}(0,1))$            ▷ search direction
        $\tilde{\mathbf{y}}_l = \mathbf{y}^{(g)} + \sigma^{(g)}\tilde{\mathbf{z}}_l$            ▷ mutate $\mathbf{y}$
        $\tilde{F}_l = F(\tilde{\mathbf{y}}_l)$            ▷ evaluate offspring
    Sort Individuals $\tilde{\mathbf{y}}$ Ascendingly w.r.t. Fitness $\tilde{F}$
    $g = g+1$
    $\mathbf{y}^{(g)} = \frac{1}{\mu}\sum_{m=1}^{\mu}\tilde{\mathbf{y}}_{m;\lambda}$            ▷ recombine the $\mu$ best $\tilde{\mathbf{y}}$
    $\mathbf{z}^{(g)} = \frac{1}{\mu}\sum_{m=1}^{\mu}\tilde{\mathbf{z}}_{m;\lambda}$            ▷ recombine the $\mu$ best $\tilde{\mathbf{z}}$
    $\mathbf{s} = (1-c)\mathbf{s} + \sqrt{\mu c(2-c)}\,\mathbf{z}^{(g)}$            ▷ update $\mathbf{s}$-path
    $\sigma^{(g)} = \sigma^{(g-1)}\exp\left(\frac{\|\mathbf{s}\|^2-N}{2DN}\right)$            ▷ update $\sigma$
until $\sigma^{(g)} < \sigma_{\text{stop}}$
```

where the parameter A denotes the oscillation amplitude and α denotes the frequency. The experiments in Section 3 were executed with $A=1$ and $\alpha=2\pi$. The global optimizer is located at $\hat{\mathbf{y}}=\mathbf{0}$. The Rastrigin function is a highly multimodal function. The number of local minima increases exponentially with the search space dimensionality N. In the case of the experiments the number of local minima is $7^{30}-1$ for $N=30$ and $7^{100}-1$ for $N=100$.

References

[1] M. Luby, A. Sinclair, and D. Zuckerman: “Optimal speedup of Las Vegas algorithms”. In: *[1993] The 2nd Israel Symposium on Theory and Computing Systems.*, 1993, p. 128–133. doi: 10.1109/TEVC.2024.3419931.

[2] J.-H., Lorenz: “Restart Strategies in a Continuous Setting”. In: *Theory of Computing Systems.*, Vol. 65. 2021, p. 1143–1164. doi: 10.1007/s00224-021-10041-0. url: https://doi.org/10.1007/s00224-021-10041-0

[3] L. Schönenberger, and H.-G. Beyer: “On a Population Sizing Model for Evolution Strategies in Multimodal Landscapes”. In: *IEEE Transactions on Evolutionary Computation.* 2024. doi: 10.1109/TEVC.2024.3419931.

[4] A. Omeradzic, and H.-G. Beyer: “Self-Adaptation of Multi-Recombinant Evolution Strategies on the Highly Multimodal Rastrigin Function”. In: *IEEE Transactions on Evolutionary Computation.* 2024. doi: 10.1109/TEVC.2024.3400857.

[5] A. Auger, and N. Hansen: “A Restart CMA Evolution Strategy with Increasing Population Size”. In: *IEEE Transactions on Evolutionary Computation.* Vol. 2. 2005, p. 1769–1776 doi: 10.1109/CEC.2005.1554902.

[6] T. Jansen: “On the Analysis of Dynamic Restart Strategies for Evolutionary Algorithms”. In: *Parallel Problem Solving from Nature — PPSN VII.* Springer Berlin Heidelberg, 2002, p. 33–43.

[7] N. Hansen: “*Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionsstrategie*”. Doctoral thesis. Technical University of Berlin, 1998.

Modalübergreifende Wissensdestillation für die Radar-Objekterkennung

Patrick Palmer, Martin Krüger und Torsten Bertram

Lehrstuhl für Regelungssystemtechnik
Technische Universität Dortmund
Otto-Hahn-Straße 8, 44227 Dortmund
E-Mail: patrick.palmer@tu-dortmund.de

1 Einleitung

Die Umgebungswahrnehmung stellt das erste Modul der Informationsprozesskette in jedem automatisierten Fahrsystem dar, wobei Sensormodalitäten wie Lidar, Kamera und Radar verwendet werden. Gegenwärtig werden vornehmlich Lidar und Kamera eingesetzt, da sie eine hochauflösende Repräsentation der Fahrzeugumgebung liefern. Lidar-Systeme sind jedoch mit hohen Kosten verbunden, während Kamerasysteme eine starke Abhängigkeit von der Umfeldausleuchtung aufweisen. Radar-Sensoren weisen diese Limitierungen nicht auf und bieten zudem Vorteile wie die direkte Messung der relativen Radialgeschwindigkeit. Die Einführung bildgebender 3+1D[1] Radar-Sensoren, welche gegenüber herkömmlichen Radaren eine höhere Auflösung erreichen, ermöglichen eine rein radarbasierte Wahrnehmung.

Alle aktuell verfügbaren Datensätze zur Radar-Objektdetektion enthalten zusätzlich zum Radar- auch einen Lidar-Sensor [1–3]. Zumeist dient der Lidar in diesen Datensätzen als Referenz, um Ground-Truth Label zu erstellen. Dies eröffnet die Möglichkeit, den Lidar im Trainingsprozess maschineller Lernverfahren einzusetzen, wobei zur Interferenz weiterhin nur der Radar-Sensor verwendet wird. Für den Transfer zwischen Sensormodalitäten werden zumeist Methoden der Wissensdestillation eingesetzt [4]. Aufgrund der identischen

---

[1] 3 räumliche Dimensionen: Entfernung, Azimut, Elevation, + relativen Radialgeschwindigkeit

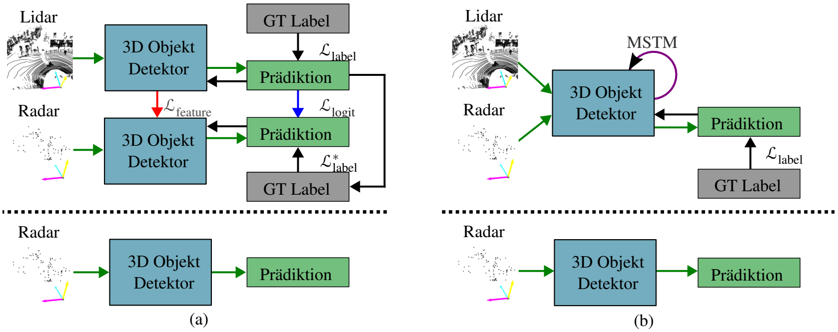


Bild 1: Überblick über die (a) auf Wissensdestillation basierte Methode und (b) die mehrstufige Trainingsmethode zur Nutzung von Lidar-Daten für das Training von Radar-Objektdetektoren. Der Trainingsprozess ist oberhalb der gestrichelten Linie, die Inferenz unterhalb dargestellt.

Datenrepräsentation als Punktwolke eröffnen sich für den Transfer zwischen Lidar- und Radar-Detektion neue Möglichkeiten, da die gleiche Basisarchitektur auf Basis neuronaler Netze verwendet werden kann.

2 Methodik

Im Rahmen dieser Arbeit werden zwei Ansätze untersucht, Wissen aus der Lidar-Objektdetektion während des Lernprozesses auf die Radar-Objektdetektion zu übertragen. Es handelt sich hierbei um einen Ansatz der Wissensdestillation (WD), welcher aus dem Bereich des rechenzeiteffizienten Netzarchitekturoptimierung abgeleitet ist, und eine mehrstufige Trainingsmethode (MSTM). Beide Ansätze sind in Bild 1 dargestellt.

2.1 Mehrstufige Trainingsmethode (MSTM)

Der MSTM-Ansatz wird wie in [5] beschrieben verwendet. Das Netzwerk wird zunächst ausschließlich auf Lidar-Daten bis zur Konvergenz trainiert. Anschließend wird eine mehrstufige Optimierung der Modellparameter durchgeführt. Die Dichte der Lidar-Punktwolke wird in jeder Optimierungsstufe halbiert. Dadurch nähert sich die Lidar-Punktwolke der Charakteristik der Radar-Punktwolke weitestgehend an; was das Netzwerk konditioniert, auf niedrig aufgelösten

Punktwolken Objekte zu detektieren. Im vorletzten Optimierungsschritt werden die Lidar- und die Radar-Punktwolke eingesetzt. Im letzten Optimierungsschritt wird dann nur noch die Radar-Punktwolke verwendet. Der MSTM-Ansatz ist schematisch in Bild 2 dargestellt. [5] hat gezeigt, dass es vorteilhaft ist die Radar-Punktwolke in jedem Trainingsschritt mit der Lidar-Punktwolke zu kombinieren, weshalb diese Kombination auch hier angewendet wird.

Für die Ausdünnung der Lidar-Punktwolke werden drei unterschiedliche Methoden untersucht. Eine zufällige Ausdünnung, eine k-nächste Nachbarn Ausdünnung, welche die Lidar-Punkte mit dem größten euklidischen Abstand zum nächsten Radar-Punkt entfernt, und eine Voxel-basierte Ausdünnung, welche die Anzahl der Lidar-Punkte in jedem Voxel zufällig reduziert, wobei jedoch eine Mindestanzahl an Punkten pro Voxel erhalten bleibt.

2.2 Wissensdestillation (WD)

Für die Wissensdestillation wird ein Detektor nur auf Lidar-Daten trainiert; dieser wird als Lehrer bezeichnet. Anschließend werden für das Training des Radar-Detektors, welcher als Schüler bezeichnet wird, unterschiedliche Kostenfunktionen $\mathcal{L}$ bezogen auf das Lidar-Modell berechnet. In dieser Arbeit werden drei Kostenfunktionen, wie in [6] beschrieben, betrachtet. $\mathcal{L}_{\text{logit}}$ und $\mathcal{L}^*_{\text{label}}$ bilden die Differenz der Lehrer und Schüler-Prädiktion. Der Unterschied beider besteht darin, dass für $\mathcal{L}^*_{\text{label}}$ nur maximale Detektionen verwendet werden. $\mathcal{L}_{\text{feat}}$ bildet die Differenz der Feature-Dimension des Lehrers und Schülers. In allen Fällen wird das Schüler-Modell zunächst mit den Parametern des Lehrer-Models initalisiert, da dies zu einer Verbesserung gegenüber einer zufälligen Initalisierung führt [6].

3 Auswertung

Alle Experimente werden auf dem View-of-Delft Datensatz [1] durchgeführt. Dieser beinhaltet einen 64-Schichten Lidar-Sensor sowie einen 3+1D hochauflösenden Radar-Sensor. Zur Evaluation wird die durchschnittliche Genauigkeit (engl. mean average precision (mAP)), wie durch [7], [8] definiert, verwendet.

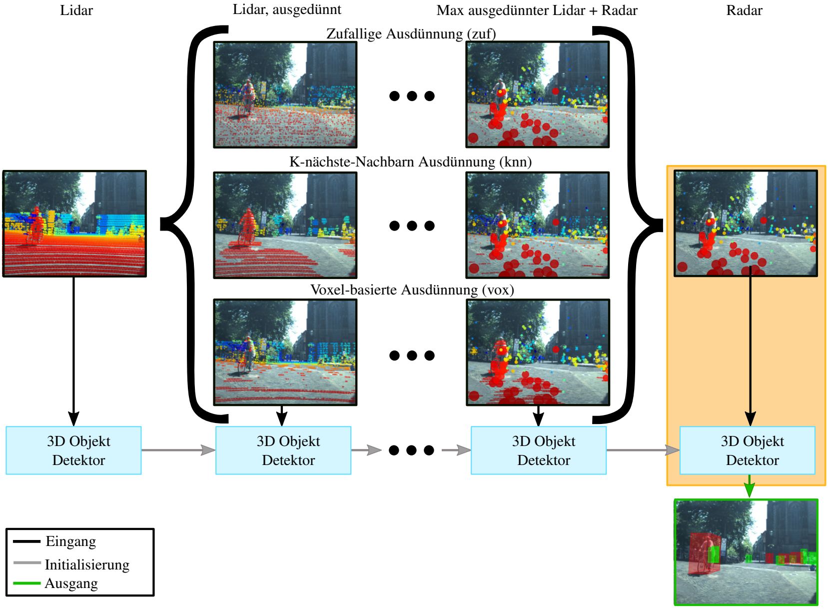


Bild 2: Schema der MSTM. Bei der Inferenz wird nur der orange schattierte Teil ausgeführt. Kleine Punkte stellen Lidar-Punkte dar, große Punkte Radar-Punkte. Die Farbe der Punkte entspricht dem Abstand zum Ego-Fahrzeug. Das Kamerabild wird nicht als Eingabe verwendet, sondern dient nur der Visualisierung.

Für eine Auswertung in unterschiedlichen Weitenbereichen wird die Auswertung in einen kurzen Messbereich (KM): 0-30 m und einen weiten Messbereich (WM): 30-50 m unterteilt [9] . Als Objektdetektor wird ein auf Radar-Daten angepasstes PointPillars [10] verwendet.

Tabelle 1, stellt die Ergebnisse der MSTM für unterschiedliche Ausdünnungsmethoden dar. Es ist zu erkennen, dass insbesondere die MSTM mit vox und knn Ausdünnung zu einer Verbesserung der Detektionsleistung führen. Wobei die Leistung von der Verkehrsteilnehmerklasse und dem Messbereich abhängt.

In Tabelle 2 sind die Ergebnisse für unterschiedliche WD-Methoden dargestellt. Zusätzlich zum Training des Lehrers auf Lidar-Daten (L) wird ein Training auf einer Punktwolke aus Radar- und einer auf ¼ der ursprünglichen Punktdichte ausgedünnten Lidar-Punktwolke ($\mathrm{RL}_{1/4\,/\,\mathrm{vox}}$) betrachtet. Es ist zu erkennen, dass alle Methoden zu einer Verbesserung im WM führen. Die Detektionsleistung, die mit einem WD-Verfahren erzielt werden kann, hängt von dem verwendeten

Tabelle 1: 3D Objektdetektionsergebnisse mit PointPillars als Detektor und der MSTM. Das beste und zweitbeste Ergebnis ist jeweils in **Fett** und Unterstrichen gekennzeichnet. R: Radar, LR: Lidar + Radar, $1\text{-}\frac{1}{16}$: Genutzte Lidar Ausdünnungsschritte $1 \rightarrow \frac{1}{2} \rightarrow \frac{1}{4} \rightarrow \frac{1}{8} \rightarrow \frac{1}{16}$.

	mAP ↑		Auto		Fußgänger		Radfahrer	
Trainingsmethode	KM	WM	KM	WM	KM	WM	KM	WM
R (Vergleichsbasis)	36,7	11,9	45,2	18,1	17,1	7,4	47,7	10,2
$\mathrm{RL}^{\mathrm{MSTM}}_{1\text{-}\frac{1}{16}\,/\,\mathrm{zuf}} \rightarrow \mathrm{R}$	35,6	14,9	44,1	20,2	17,8	7,2	44,9	**17,3**
$\mathrm{RL}^{\mathrm{MSTM}}_{1\text{-}\frac{1}{16}\,/\,\mathrm{knn}} \rightarrow \mathrm{R}$	38,2	14,7	45,5	**23,9**	**18,8**	8,2	50,3	11,9
$\mathrm{RL}^{\mathrm{MSTM}}_{1\text{-}\frac{1}{16}\,/\,\mathrm{vox}} \rightarrow \mathrm{R}$	**39,7**	**15,4**	**45,9**	22,5	18,4	**9,7**	**54,7**	13,9

Tabelle 2: 3D Objektdetektionsergebnisse mit PointPillars als Detektor und WD. Das beste Ergebnis pro WD-Methode ist **Fett** gekennzeichnet.

	Nur Init		Logit-WD		Feature-WD		Label-WD	
Lehrer-Daten	KM	WM	KM	WM	KM	WM	KM	WM
R (Vergleichsbasis)	36,7	11,9	-	-	-	-	-	-
L	34,9	13,5	32,8	12,6	34,8	13,6	**36,6**	**13,7**
$\mathrm{RL}_{\frac{1}{4}\,/\,\mathrm{vox}}$	35,2	15,3	35,6	13,3	**37,1**	**15,8**	35,3	14,3

Lehrer-Datensatz ab. Die Feature-WD profitiert von den Lehrer-Daten die strukturell nah an den Schüler-Daten sind, während die Label-WD von Lehrer-Daten profitiert die eine hohe Detektionsleistung ermöglichen.

4 Zusammenfassung

Diese Arbeit präsentiert zwei unterschiedliche Methoden, um Lidar-Daten im Trainingsprozess von Radar-Objektdetektoren zu nutzen. Die Auswertung zeigt, dass beide Methoden zu einer Verbesserung der Detektionsleistung führen. Die MSTM mit Voxel-basierter Ausdünnung führt dabei zum besseren Ergebnis und kann die Detektionsleistung um bis zu 3,5 Prozentpunkte steigern. Die Leistungsfähigkeit der MSTM hängt von der verwendeten Ausdünnungsmethode ab, während die Leistungsfähigkeit der WD-Methoden von dem verwendeten Lehrer-Datensatz bestimmt wird.

Literatur

[1] A. Palffy , E. Pool, S. Baratam, J. F. Kooij, D. M. Gavrila: „Multi-Class Road User Detection With 3+1D Radar in the View-of-Delft Dataset“. In: *IEEE Robotics and Automation Letters*. 2022.

[2] L. Zheng et al.: „TJ4DRadSet: A 4D Radar Dataset for Autonomous Driving“. In: *25th International Conference on Intelligent Transportation Systems (ITSC)*. 2022.

[3] X. Zhang et al.: „Dual Radar: A Multi-modal Dataset with Dual 4D Radar for Autononous Driving“. In: *arXiv preprint arXiv:2310.07602*. 2023.

[4] M. Klingner et al.: „X3KD: Knowledge Distillation Across Modalities, Tasks and Stages for Multi-Camera 3D Object Detection“. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

[5] P. Palmer, M. Krüger, S. Schütte, R. Altendorfer, G. Adam, T. Bertram: „LEROjD: Lidar Extended Radar-Only Object Detectiont“. In: *European Conference on Computer Vision (ECCV)*. 2024.

[6] J. Yang et al.: „Towards Efficient 3D Object Detection with Knowledge Distillation“. In: *Advances in Neural Information Processing Systems*. 2022.

[7] A. Geiger, P. Lenz, R. Urtasun: „Are we ready for autonomous driving? the kitti vision benchmark suite“. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[8] P. Palmer, M. Krüger, R. Altendorfer, G. Adam, T. Bertram: „Reviewing 3d object detectors in the context of high-resolution 3+1d radar“. In: *Workshop on 3D Vision and Robotics at the Conference on Computer Vision and Pattern Recognition*. 2023.

[9] G. Zamanakos et al.:„A comprehensive survey of lidar-based 3d object detection methods with deep learning for autonomous driving“. In: *Computers and Graphics*. 2021.

[10] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, O. Beijbom: „PointPillars: Fast Encoders for Object Detection from Point Clouds“. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

Dichte-skaliertes Optimierungskriterium für Sliced Latin Hypercube Designs

Julian Bültemeier[1]*, Marvin Schöne[2]*, Martin Kohlhase[2],
Christoph-Alexander Holst[1], Volker Lohweg[1], Oliver Nelles[3]
* gleicher Beitrag

[1]inIT – Institute Industrial IT
Technische Hochschule Ostwestfalen-Lippe,
Campusallee 6, D-32657 Lemgo, Germany
E-Mail: julian.bueltemeier@th-owl.de

[2]CfADS – Center for Applied Data Science
Hochschule Bielefeld,
Schulstraße 10, 33330 Gütersloh
E-Mail: marvin.schoene@hsbi.de

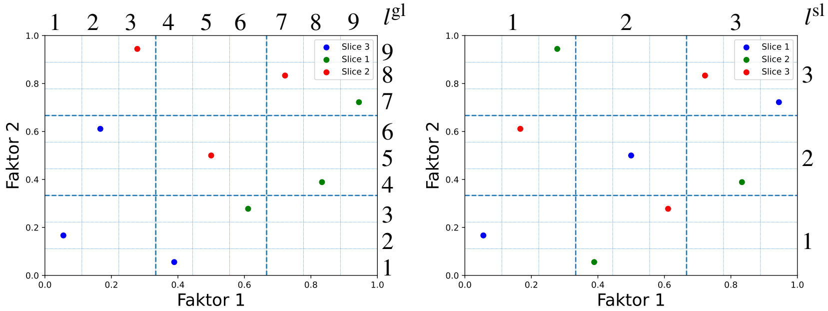
[3]Universität Siegen,
Paul-Bonatz-Str. 9-11, 57068, Siegen, Germany
E-Mail: oliver.nelles@uni-siegen.de

1 Einleitung

Die Approximation realer Prozesse durch datenbasierte Modelle erfordert in vielen Fällen die Erhebung neuer oder zusätzlicher Daten [1,2]. Dazu eignen sich Methoden des *Design of Experiments* (DoE), die auf eine effiziente Datenerhebung in Form von Versuchsplänen abzielen. Falls für den zu approximierenden Prozess keine Modellannahmen getroffen werden können, sind raumfüllende Versuchspläne vorteilhaft [3]. Etablierte raumfüllende Versuchspläne sind *Latin Hypercube Design*s (LHDs), bei denen der Unterraum jeder der M Eingangsvariablen (Faktor) in gleichverteilte Intervalle (Level) unterteilt und die N Versuchspunkte so platziert werden, dass in jedem Level l^{gl} jeder Dimension genau ein Punkt liegt (vgl. Bild 1a) [4,5].

Damit in mehrdimensionalen LHDs eine angemessene Raumfülle erzielt werden kann, ist eine Optimierung erforderlich, die häufig mit Suchalgorithmen durchgeführt wird. Diese Suchalgorithmen sind auf die Optimierung kontinuierlicher Faktoren beschränkt [6]. Jedoch müssen bei der experimentellen Untersuchung von Prozessen nicht nur kontinuierliche sondern auch kategorische Faktoren berücksichtigt werden. Letztere können beispielsweise durch eine zufällige Zuordnung eines kategorischen Faktors zu jedem Versuchspunkt berücksichtigt werden. Dadurch entsteht für jede Ausprägung des kategorischen Faktors ein sogenannter *Slice*, der eine Untermenge der Versuchspunkte beinhaltet. Bei einer willkürlichen Zuweisung des kategorischen Faktors können jedoch keine raumfüllenden Eigenschaften in den Slices und damit auch nicht für den kategorischen Faktor garantiert werden. Bild 1a veranschaulicht die daraus resultierende Problematik einer ungleichmäßigen Raumabdeckung in den Slices anhand eines zweidimensionalen Datensatzes, bei dem der Einfluss von zwei kontinuierlichen Faktoren für einen kategorischen Faktor mit 3 Ausprägungen experimentell untersucht wird.

Um dieser Problematik zu begegnen, wurden *Sliced Latin Hypercube Designs* (SLHDs) $\mathcal{D}$ entwickelt, mit denen sowohl die Versuchspunkte im Unterraum



(a) Global optimiertes Design mit zufällig zugewiesenen kategorischen Faktoren mit den N eingezeichneten globalen Level l^{gl} für beide numerischen Faktoren.

(b) Design, bei dem sowohl das globale Design als auch die Slices optimiert wurden mit den $K = \frac{N}{T}$ eingezeichneten Slice-Level l^{sl} für beide numerischen Faktoren.

Bild 1: Vergleich von 2 LHDs für die experimentelle Untersuchung des Einflusses von zwei kontinuierlichen Faktoren bei drei verschiedenen kategorischen Ausprägungen bzw. Slices. Jeder Slice in (b) besitzt Versuchspunkte, die sich über den gesamten Eingangsraum erstrecken, wohingegen die Slices in (a) nur einzelne Bereiche abdecken.

der kontinuierlichen Faktoren (globales Design) als auch die Versuchspunkte in den T Slices $\mathcal{D}_1,\ldots,\mathcal{D}_T$ optimiert werden können [7]. Hierfür wird der Eingangsraum zusätzlich in $K = \frac{N}{T}$ gleichverteilte Slice-Level l^{sl} unterteilt und die K Versuchspunkte eines jeden Slices $\mathcal{D}_t$ so platziert, dass sich in jedem Slice-Level genau ein Punkt jedes Slices befindet (vgl. Bild 1b). Dies führt dazu, dass die Versuchspunkte sowohl im globalen Design als auch in den einzelnen Slices *Latin Hypercube* (LHC)-Eigenschaften aufweisen, weshalb die Slices in Bild 1b über bessere raumfüllende Eigenschaften als in Bild 1a verfügen.

Bei der Optimierung von SLHDs müssen zwei konkurrierende Optimierungsziele in Form einer Mehrzieloptimierung (auch Pareto-Optimierung genannt) berücksichtigt werden: Die raumfüllenden Eigenschaften des globalen Designs und die raumfüllenden Eigenschaften der Slices. Bei einem Großteil der bestehenden Verfahren zur Optimierung von SLHDs wird jedoch nur das globale Design optimiert [10, 12, 14]. Andere Optimierungsverfahren vereinfachen die aufwendige Pareto-Optimierung, indem sie die beiden konkurrierenden Optimierungsziele additiv zu einem einzigen Optimierungskriterium zusammenfassen. Durch eine unangemessene Gewichtung der beiden Ziele wird jedoch das globale Design in der Optimierung stärker berücksichtigt als die Slices, was zu einem unausgewogenen Pareto-Optimum und damit zu einem suboptimalen Ergebnis für die Slices führen kann [6, 9, 11]. Der Ansatz von CHEN et al. hingegen kann vergleichbare Optima der Slices und des globalen Designs erzielen, ist aber auf SLHDs mit maximal 30 Punkten beschränkt [13]. Ein ausgewogenes Optimum kann mit den bestehenden Verfahren nur erreicht werden, wenn die Gewichtung, wie stark das globale Design und die Slices das Optimierungskriterium beeinflussen sollen, im Rahmen einer aufwendigen Pareto-Optimierung variiert wird.

Um eine Pareto-Optimierung zu vermeiden, wird in diesem Beitrag ein neues dichte-skaliertes Optimierungskriterium vorgestellt, das sowohl das globale Design als auch die Slices gleichermaßen bei einer (Einziel-)Optimierung berücksichtigt. Dieses Kriterium wird mithilfe des *Simulated Annealing* (SA) Algorithmus untersucht, der bei der Optimierung von SLHDs etabliert ist. Im Rahmen der Untersuchung wird nachgewiesen, dass eine Optimierung des dichte-skalierten Kriteriums zu einem ausgewogenen Pareto-Optimum führt.

2 Optimierte Sliced Latin Hypercube Designs

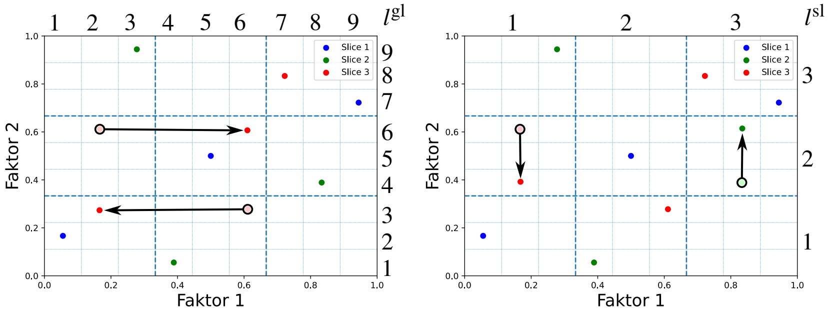
Die SLHDs ermöglichen die Erstellung von Versuchsplänen mit kategorischen Faktoren in Slices. Diese Versuchspläne bewahren raumfüllende Eigenschaften und bieten gleichzeitig die Flexibilität, sowohl numerische als auch kategorische Faktoren zu berücksichtigen. Um diese raumfüllenden Eigenschaften zu erreichen, wird eine Optimierung der SLHDs durchgeführt, die typischerweise auf dem Prinzip des Koordinatenaustauschs basiert. Dabei werden die Positionen der Versuchspunkte schrittweise so angepasst, dass ein Kriterium zur Charakterisierung der raumfüllenden Eigenschaften des globalen Designs als auch der verschiedenen Slices optimiert wird.

2.1 Erlaubte Tauschoperationen

Ein Aspekt der SLHD-Optimierung ist der Erhalt der LHC-Eigenschaften innerhalb jedes Slices. Diese Einschränkung führt dazu, dass nur bestimmte Tauschoperationen zulässig sind. Im Rahmen der Optimierung können zwei Arten von Tauschoperationen ausgeführt werden:

1. **Tausch innerhalb eines Slices**: Tauschoperationen zwischen zwei Versuchspunkten $\mathbf{x}(i)$ und $\mathbf{x}(j)$ des selben Slices haben keinen Einfluss auf die LHC-Eigenschaften des jeweiligen Slices. Infolgedessen können beliebige Wechsel von Koordinaten $x_m(i)$ und $x_m(j)$ zweier Versuchspunkte für den Faktor m durchgeführt werden, sofern $\mathbf{x}(i), \mathbf{x}(j) \in \mathcal{D}_t$ gilt. Bild 2a zeigt einen exemplarischen Tausch innerhalb des Slices 3 für den Faktor $m = 1$.

2. **Tausch zwischen zwei Slices**: Damit bei Tauschoperationen zwischen zwei Versuchspunkten unterschiedlicher Slices $\mathbf{x}(i) \in \mathcal{D}_t$ und $\mathbf{x}(j) \in \mathcal{D}_u$ die LHC-Eigenschaften der Slices bestehen bleiben, dürfen nur Koordinaten getauscht werden, die dem selben Slice-Level $l^{\mathrm{sl}} = i$ mit $i \in \{1, ..., K\}$ zugeordnet sind. In Bild 2b ist ein Tausch zwischen dem Slice 2 und 3 für Faktor $m = 2$ und dem Slice-Level $l^{\mathrm{sl}} = 2$ dargestellt.



(a) Koordinatentausch innerhalb des Slices 3 für den Faktor $m = 1$.

(b) Koordinatentausch zwischen dem Slice 2 und 3 für $m = 2$ und dem Slice-Level $l^{sl} = 2$.

Bild 2: Exemplarische Darstellung eines Koordinatentauschs innerhalb eines Slices (a) und zwischen zwei Slices (b). Die schwarzen Pfeile symbolisieren den durchgeführten Tausch.

2.2 Optimierung mit Simulated Annealing

Zur Optimierung von SLHDs wird überwiegen der SA Algorithmus [6, 14] oder eine auf SA basierende Erweiterung [8, 10–13] verwendet. SA ist ein iteratives globales Suchverfahren, bei dem zu Beginn der Suche der Lösungsraum exploriert wird. Mit fortschreitender Optimierung wird die Exploration bis zur Konvergenz der Lösung oder bis zum Erreichen einer maximalen Anzahl von Iterationen reduziert [21]. Die Optimierung erfolgt üblicherweise anhand eines kontinuierlichen distanz-basierten Kriteriums, das die raumfüllenden Eigenschaften des globalen Design und der Slices beschreibt. Algorithmus 1 illustriert einen SA Algorithmus zur iterativen Optimierung von SLHD [6]. Die Optimierung, welche im folgenden näher erläutert wird, basiert auf zufällig ausgewählten eindimensionalen Koordinatenwechseln zwischen zwei Versuchspunkten unter Berücksichtigung der in Kapitel 2.1 aufgeführten Randbedingungen.

Bevor das initiale SLHD optimiert werden kann, muss die Wahrscheinlichkeit eines Koordinatenaustausches innerhalb eines Slices $P_s \in [0,1]$ und ein Parameter zur Steuerung des Explorationsverhaltens, die Temperatur $\theta \in \mathbb{R}^+$, definiert werden. Außerdem wird in Zeile 1 der optimierte Versuchsplan $\mathcal{D}^* = \mathcal{D}$ initialisiert.

Algorithmus 1.: Simulated Annealing Algorithmus zur Optimierung von SLHDs

```
Input: Initiales SLHD: $\mathcal{D}$, Wahrscheinlichkeit innerhalb eines Slice zu tauschen: $P_{\mathrm{s}}$,
    Temperatur: $\theta$
Output: Optimiertes SLHD: $\mathcal{D}^*$
 1: Initialisiere: Optimiertes SLHD: $\mathcal{D}^* \leftarrow \mathcal{D}$
 2: While Algorithmus ist nicht konvergiert do
 3:     Zufällige Auswahl einer Dimension: $m \leftarrow \mathrm{randSel}(1,\dots M)$
 4:     If $P_{\mathrm{s}} \leq X \sim U(0,1)$ then          ▷ Koordinatentausch innerhalb eines Slice
 5:         Zufällige Auswahl eines Slice: $\mathcal{D}_{\mathrm{ex}} \leftarrow \mathrm{randSel}(\mathcal{D}_1,\dots \mathcal{D}_T)$
 6:     Else                                                 ▷ Koordinatentausch zwischen zwei Slices
 7:         Zufällige Auswahl eines Slice-Level: $k \leftarrow \mathrm{randSel}(1,\dots K)$
 8:         Selektion aller Versuchspunkte innerhalb des Slice-Level:
            $\mathcal{D}_{\mathrm{ex}} \leftarrow \{\mathbf{x}(n) \in \mathcal{D}^* \mid \frac{T(k-1)}{N} \leq x_m(n) < \frac{T\,k}{N}\}$
 9:     Zufällige Auswahl von zwei Koordinaten: $x_m(i), x_m(j) \leftarrow \mathrm{randSel}(\mathcal{D}_{\mathrm{ex}})$
10:     Koordinaten austauschen: $\mathcal{D}_{test} \leftarrow \mathrm{exchange}(x_m(i), x_m(j))$
11:     Berechnung der Wahrscheinlichkeit, ein schlechteres Design zu akzeptieren:
      $P \leftarrow \exp\left(\frac{\phi_p(\mathcal{D}^*)-\phi_p(\mathcal{D}_{test})}{\theta}\right)$
12:     If $\phi_p(\mathcal{D}_{test}) < \phi_p(\mathcal{D}^*)$ or $P \leq Z \sim U(0,1)$ then          ▷ Tausch akzeptiert
13:         Neues optimiertes SLHD akzeptieren: $\mathcal{D}^* \leftarrow \mathcal{D}_{test}$
14:     Temperatur senken: $\theta \leftarrow 0.9 \cdot \theta$
```

Für jede Iteration wird zunächst in Zeile 3 ein zufälliger Faktor bzw. eine zufällige Dimension m für den Koordinatenaustausch ausgewählt. Abhängig von der Zufallszahl $X \sim U(0,1)$ werden dann zwei mögliche Tauschpartner entweder innerhalb eines Slices für $P_{\mathrm{s}} \leq X$ (Zeile 5) oder zwischen zwei Slices für $P_{\mathrm{s}} > X$ (Zeile 7-8) ausgewählt. Der Austausch dieser Koordinaten in Zeile 10 führt zu einem neuen Versuchsplan $\mathcal{D}_{test}$, der ähnlich wie [9, 11, 13] in Abhängigkeit von einem erweiterten ϕ_p-Kriterium

$$\phi_p(\mathcal{D}) = (1-\beta)\,{\phi_p}^{\mathrm{gl}}(\mathcal{D}) + \frac{\beta}{T}\sum_{t=1}^{T} {\phi_p}^{\mathrm{sl}}(\mathcal{D}_t)\,. \tag{1}$$

und dem daraus resultierenden Optimierungsziel

$$\mathcal{D}^* = \arg\min_{\mathcal{D}} \phi_p(\mathcal{D}) \tag{2}$$

angenommen oder verworfen wird. Das ϕ_p-Kriterium ist eine kontinuierliche Approximation der minimalen Distanz [15] und wird zur Optimierung von

SLHDs aus einer Aggregation des ϕ_p-Kriterium des globalen Designs

$$\phi_p{}^{\mathrm{gl}}(\mathcal{D}) = \left(\frac{2}{N(N-1)} \sum_{\mathbf{x}(n)\in\mathcal{D}} \sum_{\mathbf{x}(j)\in\mathcal{D}\,|\,j>n} \frac{1}{d(\mathbf{x}(n),\mathbf{x}(j))^p} \right)^{p^{-1}} \tag{3}$$

und einem Mittelwert der Slices

$$\phi_p{}^{\mathrm{sl}}(\mathcal{D}_t) = \left(\frac{2}{K(K-1)} \sum_{\mathbf{x}(k)\in\mathcal{D}_t} \sum_{\mathbf{x}(j)\in\mathcal{D}_t\,|\,j>k} \frac{1}{d(\mathbf{x}(k),\mathbf{x}(j))^p} \right)^{p^{-1}} \tag{4}$$

berechnet, wobei $\beta \in [0,1]$ die beiden Terme in (1) gewichtet und $d(\mathbf{x}(n),\mathbf{x}(j))$ oder $d(\mathbf{x}(k),\mathbf{x}(j))$ der Abstand (z.B. der euklidische Abstand) zwischen zwei Versuchspunkten ist. Gemäß (2) wird der Versuchsplan als neuer optimaler Plan $\mathcal{D}^* = \mathcal{D}_{\mathrm{t}est}$ in Zeile 13 akzeptiert, wenn $\phi_p(\mathcal{D}_{\mathrm{t}est}) < \phi_p(\mathcal{D}^*)$ erfüllt ist. Damit der Algorithmus den Suchraum erkunden und lokale Minima vermeiden kann, werden mit einer Wahrscheinlichkeit von $P = \exp\left(\frac{\phi_p(\mathcal{D}^*)-\phi_p(\mathcal{D}_{\mathrm{t}est})}{\theta}\right)$ auch Versuchspläne akzeptiert, die (2) nicht weiter minimieren. Nach jeder Iteration wird die Temperatur, die die Wahrscheinlichkeit für die Annahme eines schlechteren Versuchsplans steuert, in Zeile 14 um $\theta = 0,9\cdot\theta$ verringert, so dass der Algorithmus den Suchraum weniger exploriert.

Die Optimierung wird beendet, wenn der Algorithmus ein bestimmtes Konvergenzkriterium erreicht, z. B. wenn nach einer vorab definierten Anzahl von Iterationen keine Optimierung mehr erreicht werden kann.

3 Dichte-skaliertes Optimierungskriterium

Damit für des globalen Design und die Slices zwei vergleichbare Optima erzielt werden, müssen ähnliche Änderungen im globalen Design und in den Slices auch vergleichbare Wertänderungen der zwei Summanden $\phi_p{}^{\mathrm{gl}}(\mathcal{D})$ und $\sum_{t=1}^{T} \phi_p{}^{\mathrm{sl}}(\mathcal{D}_t)$ bewirken. Unter Idealbedingungen sollten das globale Design und die Slices bei vergleichbaren raumfüllenden Eigenschaften auch die gleiche Größenordnung aufweisen. Diese Voraussetzungen werden aufgrund der folgenden zwei Effekte nicht erfüllt:

1. **Unterschiedliche Wertebereiche:** Eine Erhöhung der Anzahl der Versuchspunkte N in einem LHD bewirkt eine dichtere Verteilung der Punkte in $[0,1]^M$, was zu geringeren Abständen zwischen den Versuchspunkten führt. Dies hat zur Folge, dass abstandsbasierte Kriterien wie ϕ_p bei unterschiedlich großen Versuchsplänen mit identischen raumfüllenden Eigenschaften unterschiedliche Werte annehmen. Aus diesem Grund hat ${\phi_p}^{\mathrm{gl}}$ in (3) einen größeren Wertebereich als ${\phi_p}^{\mathrm{sl}}$ in (4).
2. **Unterschiedliche Sensitivitäten:** Ein Koordinatentausch zwischen zwei Punkten kann drastische Veränderungen des ϕ_p-Werts des globalen Designs und der ϕ_p-Werte von bis zu zwei Slices bewirken. Je mehr Slices jedoch vorhanden sind, desto weniger wirken sich diese Veränderungen auf den Mittelwert von ${\phi_p}^{\mathrm{sl}}(\mathcal{D}_t)$ in (1) aus, da sich die ϕ_p-Werte der verbleibenden Slices nicht ändern.

Folglich optimiert $\phi_p(\mathcal{D})$ in (1) das globale Design stärker als die Slices. Um dieser Unausgewogenheit entgegenzuwirken muss (4) einerseits auf eine vergleichbare Größenordnung wie (3) skaliert werden und andererseits muss der Einfluss der einzelnen Slices erhöht werden, so dass das Optimierungskriterium empfindlicher auf eine Änderung in einem Slice reagiert.

Dichte-basierte Skalierung: Damit ${\phi_p}^{\mathrm{gl}}$ und ${\phi_p}^{\mathrm{sl}}$ einen ähnlichen Wertebereich aufweisen, müssen die Versuchspunkte im globalen Design und in den Slices eine vergleichbare Dichteverteilung haben. Um eine vergleichbare Dichteverteilung zwischen den Slices und dem globalen Design zu gewährleisten, werden die Distanzen in den Slices auf einen Wert herunterskaliert, der eine mit dem globalen Design vergleichbare Punktdichte ergibt. Die Punktdichte des globalen Designs errechnet sich im Einheitswürfel aus

$$\rho_{\mathrm{g}} = \frac{N}{1} \tag{5}$$

und in den Slices aus

$$\rho_{\mathrm{s}} = \frac{\frac{N}{T}}{\alpha^M}\,, \tag{6}$$

wobei α^M das Volumen des Eingangsraums der Slices beschreibt und für eine vergleichbare Punktdichte entsprechend parametrisiert werden muss. Die

Parametrisierung für α ergibt sich aus einer Gleichsetzung von (5) und (6)

$$\alpha = \left(\frac{1}{T}\right)^{\frac{1}{M}} \tag{7}$$

und erweitert (4) durch eine Skalierung der Distanzen zu

$$\phi_p{}^{\mathrm{sl}}(\mathcal{D}_t) = \left(\frac{2}{K(K-1)} \sum_{\mathbf{x}(k)\in\mathcal{D}_t} \sum_{\mathbf{x}(j)\in\mathcal{D}_t \,|\, j>k} \frac{1}{\alpha\, d(\mathbf{x}(k),\mathbf{x}(j))^p}\right)^{p^{-1}} . \tag{8}$$

Sensitivitätserhöhung der Slices: Um die gleiche Empfindlichkeit zwischen dem globalen Design und den mit α skalierten Slices zu gewährleistet, wird zudem der ϕ_p-Wert der Slices über die Abstände aller Slices mit

$$\phi_p{}^{\mathrm{sl}}(\mathcal{D}_1,\ldots,\mathcal{D}_T)=\left(\frac{2}{N(N-1)} \sum_{t=1}^{T} \sum_{\mathbf{x}(k)\in\mathcal{D}_t} \sum_{\mathbf{x}(j)\in\mathcal{D}_t \,|\, j>k} \frac{1}{\alpha\, d(\mathbf{x}(k),\mathbf{x}(j))^p}\right)^{p^{-1}} \tag{9}$$

berechnet, statt den ϕ_p-Wert für jeden einzelnen Slice zu berechnen und den Mittelwert zu bilden.

Finales Optimierungskriterium: Basierend auf der skalierten, sensitiveren Berechnung der Slices in (9) kann das Kriterium in (1) zu einem neuen dichteskalierten Optimierungskriterium

$$\phi_p{}^{\mathrm{ds}}(\beta,\mathcal{D}) = (1-\beta)\,\phi_p{}^{\mathrm{gl}}(\mathcal{D}) + \beta\,\phi_p{}^{\mathrm{sl}}(\mathcal{D}_1,\ldots,\mathcal{D}_T) \tag{10}$$

erweitert werden, welches im folgenden hinsichtlich eines ausgewogenen Optimums evaluiert wird.

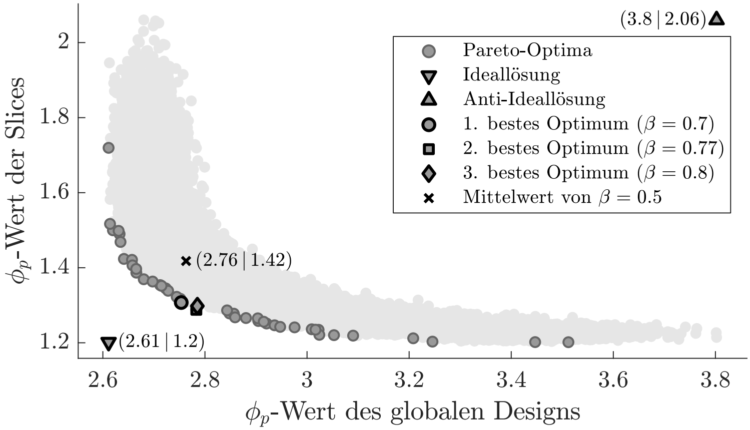
4 Evaluation

Um mit Algorithmus 1 und dem unskalierten Kriterium (1) ein ausgewogenes Optimum zu erzielen, muss β zur Gewichtung der zwei Summanden im Rahmen einer Pareto-Optimierung variiert werden. Zur Erzeugung der Pareto-Menge werden mehrere initiale SLHDs mit verschiedenen β-Werten durch

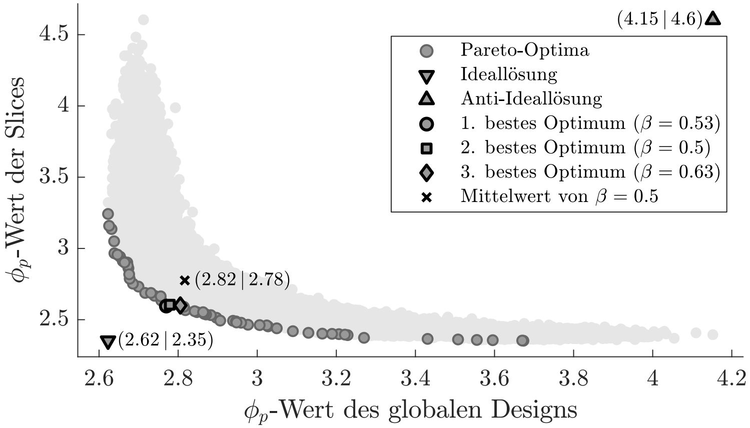
Algorithmus 1 optimiert. Die Auswahl eines Pareto-Optimums, das den besten Kompromiss zwischen den beiden konkurrierenden Optimierungszielen in Form eines ausgewogen optimalen SLHD darstellt, wird mit der Mehrziel-Entscheidungsmethode *Technique for Order Preference by Similarity to an Ideal Solution* (TOPSIS) durchgeführt. TOPSIS wählt aus der Pareto-Front die Lösung aus, die den geringsten Abstand zur Ideallösung und den größten Abstand zur Anti-Ideallösung hat. Die Ideallösung ergibt sich aus den Minima und die Anti-Ideallösung aus den Maxima der erzielbaren ϕ_p-Werte der Slices und des globalen Designs. Um repräsentative Ergebnisse bei der Auswahl einer ausgewogenen Lösung zu erzielen, werden die β-Werte der drei ausgewogensten Pareto-Optima betrachtet.

Bild 3a zeigt die Pareto-Optimierung für einen SLHD mit $N = 90$ Versuchspunkten, $T = 15$ Slices und $M = 4$ Faktoren. Zur Erzeugung der Pareto-Menge wurden 4000 initiale SLHDs durch Algorithmus 1 und (1) mit je 29 verschiedenen β-Werten im Intervall $(0,1)$ optimiert. Anhand der Pareto-Optimierung wird ersichtlich, dass die ϕ_p-Werte des globalen Designs in einem Intervall von $[2.61, 3.8]$ deutlich größere Werte annehmen können, als die ϕ_p-Werte der Slices, welche auf einen Wertebereich von [1.2, 2.06] begrenzt sind. Optimierungen, die mit $\beta = 0.5$ durchgeführt werden, führen im Mittel zu einem ϕ_p-Wert von 2.76 beim globalen Design und zu einem ϕ_p-Wert von 1.42 bei den Slices. Als die drei besten Pareto-Optima wählt TOPSIS Versuchspläne aus, die mit einen β-Wert von 0.7, 0.77 und 0.8 optimiert wurden. Daraus lässt sich schließen, dass für eine ausgeglichene Optimierung des SLHD ein β-Wert im Intervall von $[0.7, 0.8]$ verwendet werden muss. Zudem kann belegt werden, dass bei $\beta = 0.5$ das globale Design deutlich stärker optimiert wird als die Slices.

In Bild 3b ist die Pareto-Optimierung mit dem neuen dichte-skalierten Optimierungskriterium (10) dargestellt. Im Gegensatz zur Pareto-Optimierung mit dem unskalierten Kriterium weisen die ϕ_p-Werte des globalen Designs mit [2.62, 4.15] und die ϕ_p-Werte der Slices mit [2.35, 4.6] einen vergleichbareren Wertebereich auf. Eine Optimierung mit $\beta = 0.5$ erzielt im Mittel annähernd die selben ϕ_p-Werte für das globale Design (2.82) und die Slices (2.78). Ähnliche Effekte sind bei der Auswahl der drei besten Pareto-Optima zu erkennen, die mit einen β-Wert von 0.53, 0.5 und 0.63 erzielt wurden. Folglich ist mit dem



(a) Pareto-Optimierung mit dem unskalierten Optimierungskriterium in (1)



(b) Pareto-Optimierung mit dem neuen dichte-skalierten Optimierungskriterium in (10)

Bild 3: Pareto-Optimierung eines SLHD mit $N = 90$ Versuchspunkten, $T = 15$ Slices und $M = 4$ Faktoren. Auf der Abszisse ist der erzielte ϕ_p-Wert für das globale Design und auf der Ordinate der (gemittelte) ϕ_p-Wert der Slices dargestellt. Zur Erzeugung der Pareto-Menge wurden eine Vielzahl unterschiedlicher initialer SLHDs mit Algorithmus 1 und variierendem β optimiert. Mit der TOPSIS-Methode werden die drei besten und ausgewogensten Lösungen ausgewählt, um die Verzerrung des unskalierten (a) und dichte-skalierten (b) Kriteriums zu überprüfen und zu vergleichen.

dichte-skalierten Kriterium bei einem β-Wert im Intervall von $[0.5, 0.63]$ ein ausgewogen optimaler SLHD zu erwarten.

5 Zusammenfassung und Ausblick

Um einen optimalen SLHD zu erhalten, müssen sowohl das globale Design als auch die Slices optimiert werden, was eine aufwendig zu berechnende Pareto-Optimierung mit zwei konkurrierenden Optimierungszielen darstellt. Existierende Optimierungskriterien fassen diese beiden Optimierungsziele additiv zu einem einzelnen Kriterium zusammen und überführen somit die Mehrzieloptimierung in eine Einzieloptimierung. Aufgrund einer ungeeigneten Gewichtung der beiden Optimierungsziele wird jedoch das globale Design bei den existierenden Kriterien stärker optimiert als die Slices. Ein ausgewogenes Optimum kann folglich nur durch eine Pareto-Optimierung erreicht werden.

Ausgehend von einem existierenden ϕ_p-Kriterium (1) wurde in dieser Arbeit ein dichte-skaliertes Optimierungskriterium (10) entworfen und evaluiert, das ein annähernd ausgewogenes Optimum ohne die Verwendung einer aufwendigen Pareto-Optimierung ermöglicht. Dafür wurde eine dichte-basierte Skalierung der Distanzen in den Slices entwickelt, wodurch die ϕ_p-Werte des globalen Designs und der Slices vergleichbare Werte annehmen. Außerdem wurde die Empfindlichkeit des Kriteriums bezüglich Distanzänderungen in den Slices erhöht.

Für die Evaluation des neuen Kriteriums wurden dessen Summanden (3) und (9) als konkurrierende Ziele in einer Pareto-Optimierung untersucht. Dabei wurde überprüft, welche Gewichtung (β) für das neue Kriterium zur Erzielung einer ausgewogenen Optimierung notwendig ist. Die Ergebnisse der Evaluation zeigen, dass eine gleichmäßige Gewichtung des globalen Designs und der Slices ($\beta = 0.5$) zu einem annähernd ausgewogenen Optimum führt und beide Summanden ähnliche Wertebereiche bei der Optimierung aufweisen. Bei einer Pareto-Optimierung mit dem existierenden unskalierten Kriterium wurde hingegen bestätigt, dass für ein ausgewogenes Optimum die Slices mindestens im Verhältnis 7:3 gegenüber dem globalen Design gewichtet werden müssen.

In zukünftigen Arbeiten soll die Evaluation des dichte-skalierten Optimierungskriteriums weiter intensiviert werden, z. B. durch die Verwendung von Versuchsplänen mit einer anderen Anzahl an Slices, kontinuierlichen Faktoren und Versuchspunkten. Um neue Optimierungskriterien zu generieren, soll die dichte-basierte Skalierung zudem auf andere distanz-basierte Metriken angewendet werden. Darüber hinaus sollen im Rahmen eines industriellen Forschungsprojektes mit dem entwickelten dichte-skalierten Kriterium und dem hier vorgestellten SA Algorithmus reale Versuchspläne generiert werden.

Acknowledgment

Dieser Beitrag ist Teil des it's OWL Projekts *AI4ScaDa*, das vom *Ministerium für Wirtschaft, Industrie, Klimaschutz und Energie des Landes Nordrhein-Westfalen* unter der Förderkennzahl 005-2111-0015 gefördert wird.

Literatur

[1] Pfrommer, Julius and Zimmerling, Clemens and Liu, Jinzhao and Kärger, Luise and Henning, Frank and Beyerer, Jürgen „Optimisation of Manufacturing Process Parameters Using Deep Neural Networks as Surrogate Models". Procedia CIRP 2018.

[2] Kant, Girish and Sangwan, Kuldip Singh „Predictive Modelling and Optimization of Machining Parameters to Minimize Surface Roughness Using Artificial Neural Network Coupled with Genetic Algorithm". Procedia CIRP 2015.

[3] Weissman, Steven A. and Anderson, Neal G. „Design of Experiments (DoE) and Process Optimization. A Review of Recent Publications". Organic Process Research & Development 2015.

[4] McKay, M. D. and Beckman, R. J. and Conover, W. J. „Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code". Technometrics 1979.

[5] Viana, Felipe A C „Things You Wanted to Know about the Latin Hypercube Design and Were Afraid to Ask". 10th World Congress on Structural and Multidisciplinary Optimization 2013.

[6] Ba, Shan and Myers, William R. and Brenneman, William A. „Optimal Sliced Latin Hypercube Designs“. Technometrics 2015.

[7] Qian, Peter Z. G. „Sliced Latin Hypercube Designs“. Journal of the American Statistical Association 2012.

[8] Yang, Xue and Chen, Hao and Liu, Min-Qian „Resolvable Orthogonal Array-Based Uniform Sliced Latin Hypercube Designs“. Statistics & Probability Letters 2014.

[9] Zhang, Jing and Xu, Jin and Jia, Kai and Yin, Yimin and Wang, Zhengming „Optimal Sliced Latin Hypercube Designs with Slices of Arbitrary Run Sizes“. Statistics & Probability Letters 2019.

[10] Yang, Jinyu and Chen, Hao and Lin, Dennis K. J. and Liu, Min-Qian „Construction of Sliced Maximin-Orthogonal Latin Hypercube Designs“. Statistica Sinica 2016.

[11] Yuan, Ru and Guo, Bing and Liu, Min-Qian „Flexible Sliced Latin Hypercube Designs with Slices of Different Sizes“. Statistical Papers 2021.

[12] Chen, Hao and Yang, Jinyu and Lin, Dennis K.J. and Liu, Min-Qian „Sliced Latin Hypercube Designs with Both Branching and Nested Factors“. Statistics & Probability Letters 2019.

[13] Chen, Hao and Huang, Hengzhen and Lin, Dennis K. J. and Liu, Min-Qian „Uniform Sliced Latin Hypercube Designs“. Applied Stochastic Models in Business and Industry 2016.

[14] Guo, Bing and Li, Xiao-Rong and Liu, Min-Qian and Yang, Xue „Construction of Orthogonal General Sliced Latin Hypercube Designs“. Statistical Papers 2023.

[15] Johnson, M.E. and Moore, L.M. and Ylvisaker, D. „Minimax and Maximin Distance Designs“. Journal of Statistical Planning and Inference 1990.

[16] Chen, Ray-Bing and Hsieh, Dai-Ni and Hung, Ying and Wang, Weichung „Optimizing Latin Hypercube Designs by Particle Swarm“. Statistics and Computing 2013.

[17] Jin, Ruichen and Chen, Wei and Sudjianto, Agus „An Efficient Algorithm for Constructing Optimal Design of Computer Experiments“. Journal of Statistical Planning and Inference 2005.

[18] Ebert, Tobias and Fischer, Torsten and Belz, Julian and Heinz, Tim Oliver and Kampmann, Geritt and Nelles, Oliver „Extended Deterministic Local Search Algorithm for Maximin Latin Hypercube Designs“. IEEE Symposium Series on Computational Intelligence 2015.

[19] Hwang, Ching-Lai and Yoon, Kwangsun „Methods for Multiple Attribute Decision Making“. Multiple Attribute Decision Making 1981.

[20] Méndez, Máximo and Frutos, Mariano and Miguel, Fabio and Aguasca-Colomo, Ricardo „TOPSIS Decision on Approximate Pareto Fronts by Using Evolutionary Algorithms: Application to an Engineering Design Problem“. Mathematics 2020.

[21] Nelles, Oliver „Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes“. 2020.

Human-AI Co-Construction of Interpretable Predictive Models: The Case of Scoring Systems

Stefan Heid[1 †], Jaroslaw Kornowicz[2 †], Jonas Hanselle[1 3], Eyke Hüllermeier[1 3], Kirsten Thommes[2]

[1]LMU Munich
{stefan.heid,jonas.hanselle,eyke}@lmu.de

[2]Paderborn University
{jaroslaw.kornowicz,kirsten.thommes}@upb.de

[3]MCML, Munich

† equal contribution

This study explores the co-construction of probabilistic scoring systems. Using a self-developed web-based tool, called PSLVIS, participants were able to create their own decision-support models through an interactive interface. Seven academic advising experts participated, assessing the probability of student success both with and without the assistance of a Probabilistic Scoring List (PSL). The results indicate that while the co-constructed models slightly improved the experts' accuracy, they also increased decision time. Experts interacted with PSLVIS and PSL in diverse ways, displaying different levels of algorithmic aversion and appreciation. This study underscores the potential of decision-support systems that integrate data-driven algorithms with human expertise, while also revealing the wide range of challenges that need to be addressed for successful co-construction and practical implementation.

We would like to sincerely thank the participants of the study. We gratefully acknowledge funding by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG): TRR 318/1 2021 – 438445824.

1 Introduction

With the increasing access to technology and computational resources, the idea of taking advantage of Machine Learning (ML) methodology for decision support is becoming more and more feasible. Automated or partially automated decision-making with data-driven models is appealing as it can lead to more objective and accurate decisions than human decision-making alone. For example, think of decisions in the context of employee recruitment, such as hiring or placement decisions [14] in which humans alone may suffer from several biases such as “similar-to-me”-decision biases, or the data-driven construction of individualized treatment rules in personalized medicine [26].

ML models may increase the quality of decisions, but bear the problem of user acceptance: How to motivate a human decision maker to apply automated decision support systems and how to create trust and reliance in such systems [12, 13, 15]? An important prerequisite in this regard is the transparency and interpretability of the models [5,7]. Moreover, one may expect that participation, i.e., the involvement of the human expert in the process of model construction, has a positive influence, not only on acceptance [11]. Integrating humans in the process of model construction may also further improve model quality and performance — especially in cases where data is too sparse to reliably learn well-generalizing models. Hence, we introduce a *co-constructive* approach combining data-driven model induction with expert oversight.

As an underlying model class, we use so-called *scoring systems*. Roughly speaking, a scoring system proceeds from a set of (binary) features characterizing a decision context. The presence of a feature contributes a specific score (a small integer value), and a positive decision is made if the cumulative score exceeds a threshold. Models of that kind are especially comprehensible and used in many applications and fields of applied research, such as medical decision-making [18]. More specifically, we make use of PSL, an incremental and probabilistic extension of scoring systems recently developed in [10].

As a first step toward the involvement of the human expert and co-construction of a PSL, we introduce the graphical interface PSLVIS, which allows for adding, removing, and reordering features of the model as well as changing the scores.

The interface also supports the optimal (data-driven) calculation of scores and features based on the training data, thereby helping the expert to align the data with their domain knowledge. The mapping from scores to probabilities of outcomes is calculated automatically and cannot be modified. Finally, the performance of the system is visualized in the top right corner to give the user life feedback.

Building on the user interface to facilitate model co-construction, we seek to evaluate the effect of the co-constructive process on performance and reliance. More concretely, we seek to answer the following research questions:

RQ1 How does PSL influence decision-making quality compared to humans decisions without computational support?

RQ2 How do users interact with PSLVIS and navigate through the model space?

RQ3 What are the thought processes and challenges users face while using PSLVIS and applying PSL?

2 Scoring Systems and Extensions

Scoring systems are simple linear classifiers where small integer scores are assigned to each binary feature. The sum of all scores of positive features is compared against a threshold to form a decision. PSL as introduced in [10] is an extension that produces probabilistic (instead of deterministic) predictions. Moreover, it organizes the features in the form of a decision list, so that a prediction can be made at every stage. The scores of positive features are again accumulated and then mapped to a probability estimate. An example of such a stagewise model is depicted in the bottom right of Figure 1.

Scores, feature ordering, and the probability function are learned from training data. This can be achieved by starting with an empty PSL and iteratively expanding it with the most promising feature-score-pair in a greedy fashion, similar to learning decision trees. As larger total scores should yield larger probabilities, isotonic regression is employed to obtain probability estimates that are monotonically increasing in the total score. For a detailed description of the learning algorithm, we refer to [10].

At prediction time, features are evaluated one after another, updating the total score for each of them by adding up the scores of positive features. At each of these stages, the probability estimate can be looked up. If the estimate is not sufficiently informative to make a confident decision, additional features can be evaluated to refining the estimate and reduce uncertainty.

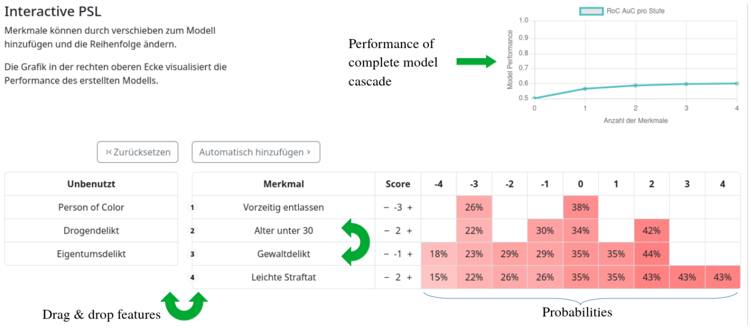
3 Co-constructive Framework: PSLvis

As a first step toward co-constructive learning of a PSL, we introduce the web interface PSLVIS instead of a purely data-driven induction. The user interface (UI) allows adding, removing, and reordering features of the model as well as changing the scores via drag-and-drop and button presses. Additionally, there are buttons to reset the model, i.e., to remove all selected features and also to add one feature optimally based on the training data. The interface also supports the optimal calculation of scores and features, allowing the experts to complement (or even replace) their expertise by a data-driven approach. The mapping from scores to probabilities is calculated automatically and cannot be modified. Finally, the performance of the entire decision list is visualized in the top right corner to give the user life feedback. A screenshot of the main view of PSLVIS is shown in Figure 1.

Significant emphasis was placed on usability during the development of the web-based UI. The UI provides an interactive experience without requiring page reloads, and any changes to features or scores result in instant model updates and performance chart adjustments. Probabilities are visually highlighted using color gradients for better clarity. The application's data model is organized into *experiments*, which can be configured independently (modifications in the user interface, different datasets, …). Participants are assigned to these experiments, and all user data is stored in an anonymized format. All UI interactions are logged in the database, enabling a detailed analysis of the co-construction process. The implementation is publicly available[1].

---

[1] `https://github.com/TRR318/pslvis`



Interactive PSL

Merkmale können durch verschieben zum Modell hinzufügen und die Reihenfolge ändern.

Die Grafik in der rechten oberen Ecke visualisiert die Performance des erstellten Modells.

Unbenutzt
Person of Color
Drogendelikt
Eigentumsdelikt

	Merkmal	Score	-4	-3	-2	-1	0	1	2	3	4
1	Vorzeitig entlassen	− -3 +		26%			38%				
2	Alter unter 30	− 2 +		22%		30%	34%		42%		
3	Gewaltdelikt	− -1 +	18%	23%	29%	29%	35%	35%	44%		
4	Leichte Straftat	− 2 +	15%	22%	26%	26%	35%	35%	43%	43%	43%

Performance of complete model cascade

Drag & drop features

Probabilities

Figure 1: User interface PSLVIS, which allows adding, removing and reordering features of the PSL via drag and drop.

4 Method

4.1 Study Dataset

The study topic chosen was student counseling, specifically focusing on assessing whether a student can successfully complete their university studies. Employees from various student counseling departments were recruited as experts for the study. The basis for the study comes from the German National Educational Panel Study [2], in which pupils and students are surveyed over a longer period. This dataset is available for research purposes. We built our dataset based on Fourage and Heß [9], where we also define dropout as whether students discontinue their initial studies at their initial institution.

In our dataset, there are a total of 1,804 students, and the success rate is 65.2%. For the study, we divided the dataset according to the participants' fields of study and only used the data relevant to the areas the participants are involved with in their work. For example, participant P1 received an engineering sample, while P4 received a sample with students from law, economics, and social sciences. The dropout rate varied slightly, and the instructions within the study explained the sample.

4.2 Think-Aloud Method

To explore how participants interact with the co-constructive tool PSLVIS and the resulting PSL, and to identify challenges encountered during their application, we employ the think-aloud method. This qualitative research method is used to elicit cognitive processes by requiring participants to verbalize their thoughts while performing tasks, with these verbalizations recorded for subsequent analysis [4, 24]. The think-aloud method serves multiple purposes, including documenting decision-making processes [19, 23] and assessing the usability and perception of products such as software [1, 8, 22, 25]. It is also increasingly utilized in human-computer interaction research [6, 16, 20, 21].

4.3 Procedure

Expert Participants. We contacted university staff with experience in academic advising and recruited seven participants. The study took place individually and in person, with participation conducted on a computer. Experimenters were present in the room, briefly explained the procedure before the start of the study, and answered any questions for clarification. All participants signed a privacy consent form before the study began. Detailed information about the participants can be found in Table 1.

Table 1: Participant Information Table

	Profession	**Major**	**Age**	**Sex**	**Sample**
P1	Study Advisor Enginering Sci.	Education	34	m	Enginering
P2	Study Advisor Enginering Sci.	Mech. Eng.	34	m	Enginering
P3	Study Advisor Enginering Sci.	Ind. Eng.	30	m	Enginering
P4	Head of Teaching/Study Center	Polical Sci.	42	m	Law/Eco./Social
P5	Study Advisor Enginering Sci.	Mech. Eng.	32	m	Enginering
P6	General Study Advisor	Education	35	f	All
P7	Study Advisor Comp. Sci.	Comp. Sci.	31	f	Math/Nat. Sci.

A) Elicitation of Mental Models. The participants' mental models regarding the decision problem are elicited. Participants rated each feature based on how they perceived the relationship between the feature and student dropout or success.

They provided a numerical rating on a scale from -100 (indicating dropout) to $+100$ (indicating success) to represent the perceived correlation.

B) Probability Assessment I. Each participant assessed the likelihood of success for 10 students. To do this, they were shown the students' features and provided a percentage-based evaluation. The 10 students were randomly selected from the eligible sample, and the order in which they were presented to each participant was randomized. No feedback was given during this stage.

C) Co-Construction with PSLvis. Participants then moved into a phase where they engaged with PSLVIS to co-construct PSL models. Their goal was to develop models that perform optimally within a constraint — the models could only expand up to five stages. This phase did not have a time limit, allowing participants to work through the process at their own pace. During this time, all interactions with the tool were logged, and participants were encouraged to verbalize their thought processes through the think-aloud method. Before the participants proceeded, the experimenters asked two questions: first, whether the participants were able to represent and encode their views in the model, and second, what the participants had focused on.

D) Probability Assessment II. In the final phase of the study, participants were asked to reassess the success probabilities of students using the PSL models they developed. This phase mirrors the initial classification task, but with the significant difference that participants could now apply their own co-constructed models. Throughout this process, the think-aloud method was employed to capture detailed insights into how participants utilize their PSL models in practice. As soon as the participants finished their second set of estimates, the experimenters asked two final questions. First, whether they had made use of the PSL levels and whether they had used all the features, and second, to what extent the PSL had influenced their decisions.

5 Results

5.1 Participants' Assessments

Table 2 presents the average times all participants took to make their assessments and their accuracy, measured by the Brier score (lower is better) [3]. The results are divided between the two assessment rounds. A purely data-driven PSL model, evaluated using individual samples for each participant, serves as the reference for accuracy.

Although a precise statistical evaluation is not possible due to the small sample size, the descriptive analysis shows that experts took longer to make their assessments in the second round. This is likely because they were also interested in reviewing their own PSL models, though there is considerable variance in this aspect. In terms of accuracy, experts generally performed slightly better with the PSL model than without, though this was not true for everyone. The reference values indicate that, on average, the experts outperformed the purely data-driven model in the second round.

Table 2: Average duration for the assessment of the students in seconds and the Brier scores (lower is better) for the first and second assessments. The PSL column serves as a reference for a purely data-driven model. The bottom row shows the average for all.

	Average Time		Brier Score		
	I	**II**	**I**	**II**	**PSL**
P1	110.8	70.8	0.29	0.28	0.26
P2	62.6	58.6	0.32	0.23	0.26
P3	75.4	94.1	0.24	0.24	0.26
P4	54.3	75.4	0.29	0.24	0.27
P5	45.2	31.6	0.33	0.26	0.26
P6	42.1	36.9	0.14	0.20	0.29
P7	19.2	104.9	0.26	0.25	0.25
ø	**58.51**	**67.46**	**0.27**	**0.24**	**0.26**

5.2 Co-construction as Navigation in the Model Space

In phase A) of Section 4.3 the participants were asked to express their mental model by providing weights for each feature in the dataset to elicit positive or negative correlation with the target class "study success". Figure 2 shows the features and the accompanying assigned scores. The features are sorted by the mean absolute score of the participant's mental model assessments, shown as blue bars. The participants assume that neuroticism is the strongest indicator for study dropout, while life satisfaction, consciousness, and openness are the three strongest indicators for study success.

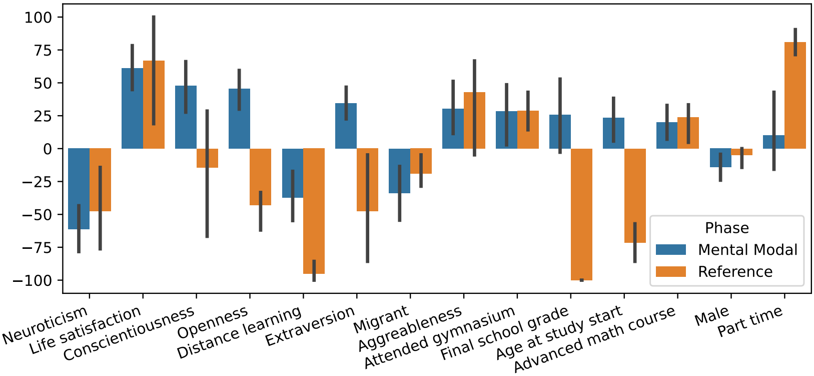


Figure 2: The blue bars show the mean feature importance assessment from phase A of the study; the orange bars show the mean score for that features when PSL is fitted on the respective data sample, normalized to the same domain $[-100, 100]$. The error bars show the 95% confidence interval of the mean.

The orange bars show the average scores of a fully data-driven PSL trained, each on the same dataset as the participant. For easier comparability, the scores from $\{-3, \ldots, +3\}$ have been rescaled to $[-100, 100]$. All dataset samples have been pooled in that figure, as the number of participants is so small. The participant's assessment of feature importance strongly disagrees with the purely data-driven feature importance as calculated from the PSL scores. In the reference model, poor final school grades, distance learning, and high age at the start of study are the strongest indicators of study dropout, while studying part-time and having high life satisfaction are the strongest indicators of success. Since the participant's goal was to have a high predictive performance on data points from this dataset, it is important to lower the model gap between the mental model and the data distribution in the domain.

During the co-construction process, features and scores can be changed. Each of these changes can be interpreted as an action that navigates from one model h to another model h' with edited features and scores. Hence, the co-constructive process can be seen as a navigation in the space of PSL models. We define the following distance function between PSL models h and h' in order to analyze how human co-constructors navigate through this space as follows:

$$d(h,h') = Kendall\big(F(h),F(h')\big) + \left\| \frac{S(h)-S(h')}{|\mathcal{S}|} \right\|,$$

which is the sum of the Kendall τ distance of the feature rankings and the L_2-norm of the normalized score difference ($\mathcal{S}$ is the set of possible scores). $F(h)$ denotes the feature ranking[2] and $S(h)$ the score assignments[3] of h.

Model changes during co-construction. The model changes during co-construction can be analyzed by comparing the current model, created by the participant, to other models. To this end, the distance between the mental model and the purely data-driven model was observed. As the mental model of part A) of the study is only observed through the feature importance scores from $[-100,+100]$, a PSL can be constructed as follows: First, the features are sorted with regard to the absolute importance score in descending order. Ties of feature importance assessments are broken arbitrarily. Second, the scores can be computed by mapping the $[-100,+100]$ interval to the score set $\{-3,\dots,+3\}$ by rescaling linearly and rounding.

Figure 3 shows the relative distance of the co-constructed model towards the mental model and the data-driven reference model over the time of the co-constructive process. All participants except P1 and P6 have an overall trend towards the data-driven model, starting with a model that is closer to their initial belief. For participants P2, P3, and P5, the final model is especially close to the data-driven model at the end of the co-construction phase. The large steps

---

[2] Features not present in h are assigned the maximum rank $|\mathcal{F}|$, with $\mathcal{F}$ being the set of all features.

[3] The score of absent features is set to 0.

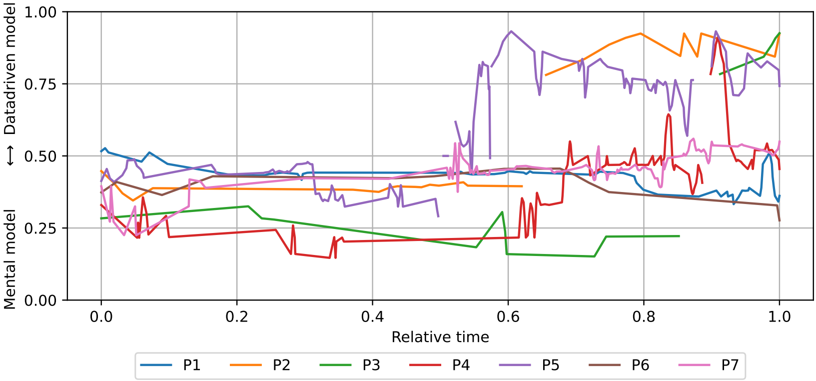


Figure 3: The relative distance between the co-constructed model for each participant towards two reference models is shown on the y-axis: one model created from the feature importance assessment ($y = 0$) and one model trained purely data-driven ($y = 1$). The x-axis shows the relative time over the course of the co-constructive process.

towards the data-driven model in P2 through P5 are caused by the participants' use of the reset and automatic feature addition buttons. However, P7 also co-constructed the model closer to the data-driven model only by manually adding features and modifying scores. When ignoring changes induced by the automatic feature addition, we can see that most participants seem to end up with models that have similar distances to their initial mental model and the data-driven reference. This is particularly illustrated with P4, where the changes from the automatic feature addition after around 90% of the co-construction time are mostly reverted manually. Similarly, P5 also modifies the model after feature addition to move closer toward their mental modal after using automatic feature addition (60%, 90% time). As Figure 3 only visualizes the relative distance to two anchor points, it still seems that most participants do not fully explore the space of models, as the relative distance changes are relatively small. Note that all co-constructive models consist of at most 5 features, while the two reference models consist of all features.

5.3 PSLvis User Actions

Figure 4 illustrates how the participants interacted with PSLVIS during the co-construction process. This is shown through a timeline for each participant, revealing several key insights: the duration of the co-construction process varied significantly. While two experts (P2, P6) spent less than 5 minutes on this part, two others (P4, P7) took more than 13 minutes.

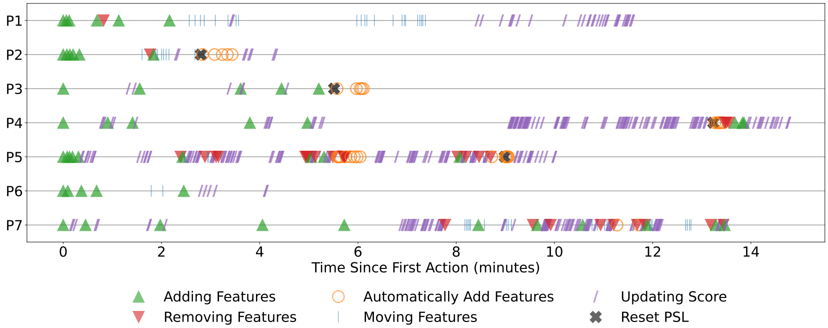


Figure 4: Action timelines for each participant, showing the time elapsed since the first recorded action. Each marker represents the subject's specific action.

All participants started by independently adding features to the model. Three of them simultaneously adjusted the scores (P3, P4, P7), while the others first focused on filling in the model. Participant P6 did not remove any features and stayed with the initially selected ones. Four participants (P2, P3, P4, P5) used the reset function, all directly related to the automatic addition of features. Of these, one participant (P3) accepted the features without adjusting the scores, two (P2, P4) only modified the scores, and one (P5) both changed the features and adjusted the scores. One expert (P7) used the automatic addition function without resetting.

5.4 Think-Aloud Results

The audio data was first transcribed and then inductively coded after multiple readings. First, the data was categorized into statements about PSLVIS and/or PSL, and second, into statements about thought processes and/or challenges.

5.5 Co-Construction with PLSvis

Thought Processes. Participants in the co-construction process with PSLVIS engage in various strategies as they explore and modify the model. They add features they believe are important, sometimes based on their intuition or domain knowledge. However, they also experiment with different feature combinations and observe how these changes impact the model's performance: *"I'll throw in what I think might be important. Maybe I can also just throw in a lot and delete it afterward."* (P1). Performance is constantly evaluated, and features are removed if they do not contribute positively to the results. In some cases, participants experiment with features even if they do not fully understand them (e.g., the "Life Satisfaction" feature) just to observe how the performance changes.

Additionally, scores are tested to understand their influence on the performance: *"I can still tweak the scores a bit, but no matter what changes I make, the model performance always gets worse."* (P2)

The tool's ability to automatically suggest features is also tested, and while these suggestions may not always align with the participant's intuition, they may still be retained: *"I wanted something to be added automatically, and then it gave me 'Agreeableness'. That's a trait I haven't thought much about, but it can certainly make sense."* (P7). Throughout the process, participants remain mindful of the five-feature limit, which shapes their decisions about feature inclusion and removal: *"I would have liked to add more than five traits, but I'm not sure if that had made it more accurate."* (P7).

Challenges. Several challenges emerged during the co-construction process. An expert encountered features that are rare in practice, such as "Part-Time Studies While Working," which created confusion about their relevance: *"I actually noticed during the modeling process that I disagreed with at least* *one selection of traits, because it was about a part-time study program. If I remember correctly, none of the students were actually studying part-time. That was a trait I only included because it significantly improved the model's performance. In hindsight, I think I would choose against it. This means I definitely didn't blindly follow the model, because I noticed this issue while working with it."* (P7).

Additionally, problems arose when thresholds led to scores that appeared counterintuitive, causing frustration as the participants struggled to understand why a certain threshold resulted in an "unnatural" decision boundary.

One expert expressed a desire to revise their models during the second estimation phase: *"You can't go back. Damn! I should have... Ugh, crap. I should have actually given a minus point for 'Migrant'."* (P1). There were also concerns about model performance, with some participants perceiving the performance as suboptimal. Many felt that the limit of five features was too restrictive for building effective models: *"It's incredibly difficult now with these five things I've chosen. I do believe that they are all relevant, but so is the rest. At least in part."* (P4).

For an expert, it is not clear how high or low the scores can be set (presumably due to the previous example explanation, where the scores only went up to $+2$): *"I'll play around a bit with the scores. I can do them too. I somehow thought I could only make it up to plus and -2, but I can make them up to seven. That's relevant, of course."* (P7).

Some experts noted discrepancies between the data provided and their real-world experiences, further diminishing confidence in the tool: *"Uh, difficult. I generally found it challenging to align my experience from my specialized counseling sessions with the traits you have. So, the selection of traits wasn't really good. I would rarely classify my counseling sessions based on what you have."* (P5). Challenges also arose with binary features; for example, when a student was female, participants found it unclear how to use the feature 'Male'. Finally, the direction of certain features, such as 'Final school grade,' created confusion, as the relationship between the feature and the score did not always align with the participant's expectations.

5.6 Decision-Making with PSL

Thought Processes. When using PSL, experts tend to go through the process methodically, often calculating probabilities all the way to the end. They adjust the output on occasion, but not always; in some cases, they accept the PSL-generated probability as is. One reason for adjusting the output was that the expert had a different weighting of features in mind compared to the system: *"Okay, I tried it with the model, and it would be 62%. When I think about it now: 18 years old, relatively young, 2.5 final grade — let's say an average school diploma. Male. Not a migrant, took advanced math courses in school. (...) Yeah, I can see again in my own evaluation that, as I said, I tend to rate all these soft skills or personality traits lower than I probably should."* (P3).

PSL influenced the estimation behavior of the participants. One expert noted that they felt motivated to deviate more from the average value when they saw the PSL probabilities, suggesting that the tool impacted their decision-making strategy: *"And if the model now gives me 86%, I'm actually more motivated, let's say, to deviate a bit more from this average score than before. So, I'll go with 75%."* (P3).

Some participants were not concerned about small differences in probabilities; minor variations did not affect their overall judgment: *"In the end, it doesn't really matter whether someone has a 75% or 85% probability of success. But it definitely makes a difference whether they have 40% or 75%."* (P4)

Challenges. One notable challenge with PSL was the inability to modify the model during the second estimation phase. Another challenge arose from the fact that a 0% probability is practically impossible in real-world scenarios. For one expert, receiving such a result led to significant aversion: *"The probability of successfully completing an engineering degree will never be 0%, because, well, if you have enough people, someone will always manage to do it. So, in this case, I would deviate significantly from the model and estimate it around 60%."* (P3).

6 Discussion

In this study, we focused on the interactive co-construction of interpretable predictive models, specifically through the lens of probabilistic scoring systems. To this end, we developed a web-based user interface that allows experts to construct their own PSL models and co-construct them with the PSL model. In a study involving 7 experts, we investigated how PSL influences the decision-making quality of users, how the experts co-construct their models, and how the interaction unfolds, identifying where challenges arise.

First, the results show that co-construction can slightly improve experts' performance in terms of accuracy, although at the cost of longer decision times. Notably, the co-constructed models also outperformed purely data-driven models. While we expected performance improvements due to co-construction and anticipated longer decision times due to the interpretability and computational complexity of PSL, the slightly better performance compared to the data-driven model can be explained by the complexity of the decision problem and the limited dataset. This also highlights that co-construction can offer an advantage, though this was not the case for all participants.

It is also important to note that there were different forms of co-construction. Some participants shifted from their own mental models towards the data-driven model, while others were resistant to the automated assistance [7]. This was evident in the think-aloud results: experts initially relied on their own opinions but experimented with different combinations of features and scores, occasionally guided by the automated function, even if they did not fully understand it. This corresponds to the issue of over-reliance or automation bias, often observed in human-AI interactions [17]. Participants partially relied on the PSL, not blindly, but taking it as advice that influenced their own judgment. However, there was aversion when the advice deviated too much or seemed unrealistic.

Our study also highlights challenges that can arise in human-AI interaction research, which may not be immediately apparent to researchers during development. For example, difficulties in understanding feature thresholds or the binary nature of features, especially when the data in experiments does not match real-world practice.

7 Limitations and Future Research

A key limitation is the small number of participants, preventing statistical analysis of how PSL impacted expert decisions. This is common in human-computer interaction research with experts. Future studies might consider using laypeople via platforms like Prolific, requiring familiar datasets and problems. Although not experts, a larger sample would be more cost-effective.

Another issue is the dataset used. Estimating academic success and dropout rates is complex, and the available data was limited, resulting in low model accuracy and minimal expert improvement. Future studies could benefit from better data to enhance model performance and highlight interaction effects.

Additionally, this study did not explore how participants handle missing information during decision-making, a key focus of PSL. We kept all information available to simplify the decision problem. Future research could examine how participants manage missing data or time pressure, where they have all the information but limited time to assess everything, possibly requiring more experience with PSL.

This study highlights both the potential advantages and the challenges of co-constructed and interpretable machine learning models in decision support. While the results suggest that models created by experts can slightly improve the accuracy of their decisions, they also require significantly more time for decision-making. The co-constructive interaction with the web-based tool we developed was highly varied in terms of how the functionalities were used and experimented with, as well as in the adoption of algorithmic suggestions and the adaptation of models to individual mental models. However, some issues should be addressed in future research.

References

[1] Obead Alhadreti and Pam Mayhew. Rethinking thinking aloud: A comparison of three think-aloud protocols. *Proceedings of Conference on Human Factors in Computing Systems*, 1-12, 2018.

[2] Hans-Peter Blossfeld, Hans-Günter Roßbach, and Jutta von Maurice. The German National Educational Panel Study (NEPS). *Zeitschrift für Erziehungswissenschaft: Sonderheft*, 14, 2011.

[3] Glenn W. Brier. Verificiation of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1-3, 1950.

[4] Elizabeth Charters. The use of think-aloud methods in qualitative research an introduction to think-aloud methods. *Brock Education Journal*, 12(2):68–82, 2003.

[5] Lingwei Cheng and Alexandra Chouldechova. Overcoming algorithm aversion: A comparison between process and outcome control. *Proceedings of Conference on Human Factors in Computing Systems*, 1-27, 2023.

[6] Michael Chromik, Malin Eiband, Felicitas Buchner, et al. I think I get your point, AI! The illusion of explanatory depth in explainable AI. *Int. Conference on Intelligent User Interfaces*, 26, 307–317, 2021.

[7] Berkeley J. Dietvorst, Joseph P. Simmons, and Cade Massey. Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them. *Management Science*, 64(3):1155–1170, 2018.

[8] Mingming Fan, Yiwen Wang, Yuni Xie, et al. Understanding how older adults comprehend COVID-19 interactive visualizations via think-aloud protocol. *Int. Journal of Human–Computer Interaction*, 39(8):1626–1642, 2023.

[9] Didier Fouarge and Pascal Heß. Preference-choice mismatch and university dropout. *Labour Economics*, 83, 102405, 2023.

[10] Jonas Hanselle, Johannes Fürnkranz, and Eyke Hüllermeier. Probabilistic scoring lists for interpretable machine learning. *Proceedings of DS 23rd International Conference on Discovery Science*, Springer: 189-203, 2023.

[11] Jaroslaw Kornowicz and Kirsten Thommes. Algorithm, expert, or both? Evaluating the role of feature selection methods on user preferences and reliance. *arXiv:2408.01171*, 2024.

[12] Olesja Lammert, Birte Richter, Christian Schütze, et al. Humans in XAI: increased reliance in decision-making under uncertainty by using explanation strategies. *Frontiers in Behavioral Economics*, 3: 1377075, 2024.

[13] Jörg Papenkordt, Axel-Cyrille Ngonga Ngomo, and Kirsten Thommes. Are numbers or words the key to user reliance on AI? *Academy of Management Proceedings*, Vol. 2023, No. 1: 12946, 2023.

[14] D. Pessach, G. Singer, D. Avrahamia, et al. Employees recruitment: A prescriptive analytics approach via machine learning and mathematical programming. *Decision Support Systems*, 134: 113290, 2020.

[15] Tobias M. Peters and Roel W. Visser. The importance of distrust in AI. In Luca Longo, editor, *Explainable Artificial Intelligence - First World Conference*, xAI 2023, Lisbon, Portugal, July 26-28, 2023, Proceedings, Part III, volume 1903 of Communications in Computer and Information Science: 301-–317. Springer, 2023.

[16] Snehal Prabhudesai, Leyao Yang, Sumit Asthana, et al. Understanding uncertainty: How lay decision-makers perceive and interpret uncertainty in human-AI decision making. *Proceedings of Int. Conference on Intelligent User Interfaces*, 28:379–396, 2023.

[17] Max Schemmer, Niklas Kuehl, Carina Benz, et al. Appropriate reliance on AI advice: Conceptualization and the effect of explanations. *Proceedings of Int. Conference on Intelligent User Interfaces*, 28:410–422, 2023.

[18] Albert J. Six, Barbra E. Backus, and Johannes C. Kelder. Chest pain in the emergency room: value of the heart score. *Netherlands Heart Journal*, 16:191–196, 2008.

[19] Paul Solomon. The think aloud method: A practical guide to modelling cognitive processes. *Information Processing & Management*, 31(6):906–907, 1995.

[20] Richard Stromer, Oskar Triebe, Chad Zanocco, and Ram Rajagopal. Designing forecasting software for forecast users: Empowering non-experts to create and understand their own forecasts, *arXiv:2404.14575*, 2024.

[21] Geletaw S. Tegenaw, Demisew Amenu, Girum Ketema, et al. Evaluating a clinical decision support point of care instrument in low resource setting. *BMC Medical Informatics and Decision Making*, 23(1): 51, 2023.

[22] Thomas Van Gemert, Kasper Hornbæk, Jarrod Knibbe, and Joanna Bergström. Towards a bedder future: A study of using virtual reality while lying down. *Proceedings of Conference on Human Factors in Computing Systems*: 1–18, 2023.

[23] Jacqueline Whalley, Amber Settle, and Andrew Luxton-Reilly. A think-aloud study of novice debugging. *ACM Transactions on Computing Education*, 23(2):1–38, 2023.

[24] Michael D. Wolcott and Nikki G. Lobczowski. Using cognitive interviews and think-aloud protocols to understand thought processes. *Currents in Pharmacy Teaching and Learning*, 13(2):181–188, 2021.

[25] Xuesong Zhang and Adalberto L. Simeone. Using the think aloud protocol in an immersive virtual reality evaluation of a virtual twin. *Proceedings of Symposium on Spatial User Interaction*: 181–188, 2022.

[26] Y. Zhao, D. Zeng, A.J. Rush, and M.R. Kosorok. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118, 2012.

Dieser Tagungsband enthält die Beiträge des 33. Workshops „Computational Intelligence“ der vom 21.11. – 22.11.2024 in Berlin stattfindet.

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.

ISBN 978-3-7315-1388-9
9 783731 513889

Gedruckt auf FSC-zertifiziertem Papier