

SCHRIFTENREIHE DES INSTITUTS FÜR
ANGEWANDTE INFORMATIK | AUTOMATISIERUNGSTECHNIK
KARLSRUHER INSTITUT FÜR TECHNOLOGIE (KIT)

Band 54

Herausgeber

F. HOFFMANN

E. HÜLLERMEIER



Dortmund | 26. – 27. November 2015

PROCEEDINGS **25. WORKSHOP**
COMPUTATIONAL INTELLIGENCE

F. Hoffmann, E. Hüllermeier (Hrsg.)

Proceedings. 25. Workshop Computational Intelligence

Dortmund, 26. – 27. November 2015

Schriftenreihe des
Instituts für Angewandte Informatik / Automatisierungstechnik
am Karlsruher Institut für Technologie
Band 54

Eine Übersicht aller bisher in dieser Schriftenreihe erschienenen Bände
finden Sie am Ende des Buches.

PROCEEDINGS **25. WORKSHOP**
COMPUTATIONAL INTELLIGENCE

Dortmund, 26. – 27. November 2015

F. Hoffmann
E. Hüllermeier
(Hrsg.)

Impressum



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

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

www.ksp.kit.edu



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



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

Print on Demand 2015

ISSN 1614-5267

ISBN 978-3-7315-0432-0

DOI: 10.5445/KSP/1000049620

Inhaltsverzeichnis

B. Schott, J. Stegmaier, M. Takamiya, R. Mikut (KIT Karlsruhe) Challenges of Integrating A Priori Information Efficiently in the Discovery of Spatio-Temporal Objects in Large Databases	1
S. Torkamani, V. Lohweg (inIT) Shift-Invariant Feature Extraction for Time-Series Motif Discovery	23
D. A. Menges, S. Wessing, G. Rudolph (TU Dortmund) Asynchrone Parallelisierung des SMS-EMOA zur Parameteroptimierung von mobilen Robotern	47
M. Wahby, M. Divband, S. v. Mammen, H. Hamann (Univ. of Paderborn, Univ. of Augsburg) Evolution of Controllers for Robot-Plant Bio-Hybrids: A Simple Case Study Using a Model of Plant Growth and Motion	67
S. Bagheri, W. Konen, C. Foussette, P. Krause, T. Bäck, P. Koch (Cologne University of Applied Sciences, divis GmbH, Leiden University) SACOBRA: Self-Adjusting Constrained Black-Box Optimization with RBF	87
S. Lu, E. Hüllermeier (Universität Paderborn) Locally weighted regression through data imprecisiation	97

S. Uttendorf, L. Overmeyer (Leibniz Universität Hannover) Fuzzybasierte Auslegung von Wegenetzen für fahrerlose Transportsysteme	105
H. Schulte (HTW Berlin) Ein Beitrag zur robusten Beobachtersynthese für Takagi-Sugeno Modelle mit nichtmessbaren Prämissenvariablen	121
T. Loose (HAW Heilbronn) Benchmark-Anwendung zur Schwingungsanalyse und -dämpfung von Regalbediengeräten am Beispiel eines Labormodells	127
M. Gringard, A. Kroll (Uni Kassel) Zur Homogenisierung von Breitbandtestsignalen für die nichtlineare Systemidentifikation am Beispiel eines elektromechanischen Stellantriebs	145
W. Doneit, R. Mikut, L. Gröll, M. Reischl (Karlsruher Institut für Technologie) Vorwissen in Funktionsapproximationen durch Support-Vektor-Regression bei schlechter Datenqualität	163
J. Schneider, F. Igelbrink, W. Brockmann (Universität Osnabrück) Inkrementelle nichtlineare Dimensionsreduktion	183
J. H. Schoenke, W. Brockmann (Universität Osnabrück) Approximatoren für On-line Regression	199
M. Bator, S. Torkamani, V. Lohweg (inIT) Relevanzbewertung technischer Informationen mittels Data-Mining Verfahren	217

- R. Ewerth, A. Balz, J. Gehlhaar, K. Dembczyński, E. Hüllermeier** 235
(Jena University of Applied Sciences, University of Marburg,
Technical University of Poznan, University of Paderborn)
Depth Estimation in Monocular Images: Quantitative versus
Qualitative Approaches
- M. Oeljeklaus, F. Hoffmann, T. Bertram** 241
(TU Dortmund)
Convolutional Neural Networks für die semantische
Segmentierung von Szenen
- A. Fischbach, J. Stork, M. Zaefferer, S. Krey, T. Bartz-Beielstein** 255
(TU Köln)
Analyzing Capabilities of Latin Hypercube Designs Compared
to Classical Experimental Design Methods
- J. Belz, O. Nelles** 271
(University of Siegen)
Function Generator Application:
Shall Corners be Measured?
- A. Cavaterra, T. Aissa, S. Lambeck** 289
(Hochschule Fulda)
Vergleich von Syntheseverfahren für Takagi-Sugeno Fuzzy
Systeme anhand realer Raumklima-Daten

Challenges of Integrating A Priori Information Efficiently in the Discovery of Spatio-Temporal Objects in Large Databases

Benjamin Schott¹, Johannes Stegmaier¹,
Masanari Takamiya², Ralf Mikut¹

¹ Karlsruhe Institute of Technology, Institute for Applied Computer Science,

² Karlsruhe Institute of Technology, Institute of Toxicology and Genetics
D-76344 Eggenstein-Leopoldshafen, Hermann-von-Helmholtz-Platz 1
E-Mail: benjamin.schott@kit.edu

Motivation

Current research in areas such as life sciences, energy, developmental biology, traffic management and further highly actual fields often relies on the analysis of large amounts of spatio-temporal 2D+t and 3D+t data [1].

New developments in tracking technologies in combination with recent data acquisition techniques enable the collection of a tremendous amount of time resolved object localization data represented as trajectories in an unprecedentedly high resolution [2, 3]. Such collections of comprehensive object data enable the description of movement behavior and dynamics of the underlying objects. The traditional way of analyzing such object data is by visual inspection of human observers [4]. However, the tremendous amount of data yields to the problem that whole datasets can hardly be evaluated and analyzed manually [3]. Therefore sophisticated data analysis solutions are required (see therefore [5]). In significant applications in various domains (e.g. analyzing traffic datasets) such analysis solutions are required to extract information out of the collected data (e.g. identify and utilize groups of objects that exhibit similar patterns). The extracted information serves as a basis to gain knowledge about domain-specific object characteristics [6]. This is the basis for a deeper understanding of object behavior and underlying processes. To support this knowledge discovery process [7], different visual visualization can support the understanding of

the analysis task and may serve as a feedback for the user [8]. There is a wide range of applications that require the detailed analysis of 2D+t and 3D+t object data. Due to advances in telecommunications, GPS technology, robotics, Web traffic monitoring and computer vision applications large amounts of spatio-temporal object data become more and more available [9]. The field of applications ranges from traffic management, surveillance and security [10, 11], airspace monitoring [12] to animal behavior, human movement [13], developmental biology [14] and social behavior studies [15] as well as sport scene analysis [6], to name only a few. The aim of such applications is, for example, the analysis of atmospheric phenomena and climate change, the optimization of transportation routes, a more detailed understanding of biological events in model organisms [14, 9], as well as the monitoring, modeling and simulation of dynamic phenomena [12, 16] and the improvement of tracking performance [17]. The large amounts of spatio-temporal data within the different types of applications introduce new challenges to data analysis and include the increasing need for novel approaches for the knowledge discovery process [18].

Regarding the analysis task, the degree of (dis)-similarity between trajectories is assessed using specific distance functions. On the basis of these trajectory similarities, often automated clustering methods are used to access the problem of gaining knowledge about the objects [19]. In the context of trajectory object data, clustering groups of trajectories based on their geometric proximity in spatial, temporal or spatio-temporal space. Here, the clustering can be based either on the raw trajectory data, extracted trajectory features or on models [20]. Examples of the different clustering techniques can be found in [19, 21, 22, 23]. The concept of multi-step clustering extends the simple one-layer clustering approach to multiple layers [24]. Here clustering can be applied in each layer to assess the characteristics of spatio-temporal objects more precisely. To cope with highly heterogeneous spatio-temporal objects, fully automated multi-step clustering approaches [25, 10] are not adequate enough. Therefore, interactive multi-step clustering approaches are introduced in [26, 27] to implement the application-specific *a priori* knowledge. In each step of such a cluster framework the user can choose different characteristics of objects to focus on. So it is possible to gradually build a comprehensive understanding of the set of complex objects [24, 26]. The current research therefore focuses on the combination of using human knowledge and apply-

ing computer-based methods, such as clustering, to access the underlying information of objects.

However, the quantity and quality of *a priori* information varies according to the level of detail of the considered objects. Primarily on very high level of detail the quantity and quality of the *a priori* information is not sufficient enough to serve as a basis for semi-supervised methods such as clustering. Furthermore, depending on the focus of the analysis, a different validity of allocating information to the objects is required (e.g. only very rough trends in groups of objects on a lower level of detail are required, or on the contrary a very precise allocation of information on a high level of detail is needed).

To handle this problem we present a comprehensive framework starting with the raw object data over object detection and tracking up to the knowledge discovery process to extract valuable information. Due to the complexity of the heterogeneous behavior of objects, the human analyst can interact at various levels of detail in different ways (clustering, labeling, etc.) with the dataset to extract, analyze and visualize desired information directly from the interactively selected objects.

The remainder of this paper describes the framework in detail and specifies out the interaction interfaces. Next to the introduction of the framework, one possible application of interactive knowledge discovery is presented. The validation of the presented methods is performed using a benchmark dataset modelled on the basis of real biological data [31].

Methods

In this section the general framework for designing the knowledge discovery process is described in general. Further the knowledge discovery process itself is explained more in detail.

General Framework - Design the Knowledge Discovery-Process

A general scheme for designing the knowledge discovery (KD) process is shown in Figure 1. The aim is to generate a knowledge database containing

all relevant information about the object database to answer a scientific question. To achieve this aim, the KD-process is designed iteratively to access the information within the object database. Due to the fact that *a priori* information often exists in different levels of detail within the object database, this stepwise approach leads to a very effective integration of the *a priori* knowledge. This is caused by the fact, that knowledge often appears during the KD-process on the basis of interim results. In contrary it would also be often possible to design the KD-process in only one step. However, creating such an one-step approach accessing all properties requires an enormous effort, irrespective of the interpretability. Further it is often not possible to integrate *a priori* knowledge efficient by such a one-step approach. Moreover, in each step of the framework, the problem formulation task serves as a basis for guiding the computational power towards the extraction of the required information. To support the interaction between user and framework, evaluation and visualization techniques serve as a feedback for optimizing the KD-process. In general the scheme can easily be extended and modified due to the modular architecture of the concept. The visualization techniques and the KD-process have further to be adapted to the actual application (e.g. to object classification and clustering in spatio-temporal datasets).

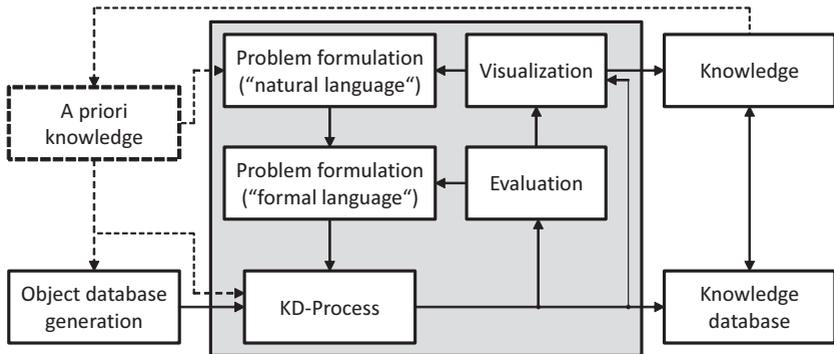


Figure 1: Scheme for the design of a knowledge discovery process. Here the effective integration of *a priori* knowledge to create a knowledge database is shown. Modified from [32].

Object Database Generation

In general an object database contains the data of several objects. In our case, the object database contains spatio-temporal object data. This serves as a basis for the KD-process for generating a knowledge database. There are many possible approaches to create such an object database, e.g. generate tracks of objects on the basis of raw object data, generate a benchmark on the basis of the raw object data or generate a data-independent benchmark (artificial benchmark, unrelated to any real world data), to name only a few. This processing step is modular and extensible, so further approaches to generate the object database can be added easily. In real world tracking projects however the tracks are not always complete over the full time period. This leads to artifacts based on short tracks. In this paper we assume a preprocessing to handle such problems.

Problem Formulation

Within each iteration of the KD-process, the first step is to specify the aims and restrictions, to identify an appropriate problem formulation in a natural language (e.g. find groups of objects that exhibit specific movement patterns). Based on this result, the natural language description of the problem has to be translated from the engineer into a more formal language (e.g. extract specific features of the objects), to be used within the algorithms of the framework. Such an interface, connecting both problem formulations, is fundamental to KD-process. This is an important prerequisite for an effective execution of the KD-process to extract meaningful knowledge out of the object databases.

Evaluation and Visualization

The evaluation and visualization of results in several processing steps serve as a feedback for the user. According to this feedback, the user is able to interact appropriately and to optimize the outcome of the process. The visualization is an important component because humans cannot deal well with large amounts of spatio-temporal object data. Therefore, visual rep-

representations are essential to support the human perception and reasoning process. This serves as a basis to assist the human analyst in understanding the object database.

To quantify the gathered results, the evaluation provides the human analyst with statistics of characteristics for groups of objects. Here, groups of objects can be evaluated according to selected spatial, temporal, spatio-temporal or object-based characteristics (e.g. color coding according features). This serves as a basis to compare several groups of objects and to derive statements about their behavior. This information can be used as additional feedback for the user to guide the KD-process.

Knowledge Discovery Process

The knowledge discovery process itself is shown in Figure 2. It consists of an information allocation process and the possibility for interactive modifications. For each step within the overall framework (Figure 1), there can be many iterations of information allocation and interactive modifications within the KD-process. Also the KD-process has a modular structure with standardized interfaces. This allows the modification and integration of methods within the process. Visualization and evaluation techniques serve again as a feedback for the user.

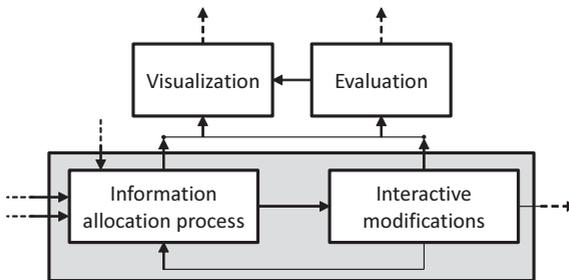


Figure 2: Scheme for the KD-process embedded within the general knowledge discovery framework in Figure 1.

Information Allocation Process

In this processing step, information is assigned to every object within the database. This can be done by extracting characteristics for each object or by manually assigning specific information to each object. Based on the assigned information, the objects can be grouped according to similar characteristics of information. In general, there are different methods to allocate the information. In the following, three basic methods for information allocation (cluster analysis, interactive filtering and manual information allocation) are described more in detail.

In cluster analysis, objects are grouped according to predefined characteristics, either in spatial, spatio-temporal or other feature spaces, in such a way that objects belonging to the same group are more similar to each other compared to those objects belonging to other groups. Based on the extracted characteristics, the information is allocated to each object as the membership to a specific group. For the clustering method, there exist several approaches that can be chosen according to the underlying problem formulation, e.g. density-based clustering, hierarchical clustering, partitional clustering, fuzzy clustering as well as continuous online clustering approaches, to name only a few [19, 28, 29]. Furthermore all parameters of cluster analysis, like the similarity measure, the extracted characteristics and the cluster method itself are modular and can be combined within the framework. Additionally, new components can easily be added and implemented in the framework. This allows a very flexible and efficient investigation of the problem.

However, when *a priori* knowledge is vague and cannot be fully described through clear characteristics, an automated approach for information allocation, such as clustering, may be not appropriate anymore (e.g. there are no appropriate features available). This often happens on very detailed object levels suffering from low quality of the underlying *a priori* knowledge about expected or interesting results. In this case, the information can be directly assigned to the objects manually. In general, manual information allocation is an appropriate method applied to a detailed object level, containing only a few objects. To make the manual allocation process more effective, a visual framework is designed to support the user. Additionally, features or combinations of features that describe the manually labeled data the best can be extracted and used for further analysis.

The third method for information allocation is interactive filtering. In this approach parameters for filters can be set manually, including temporal filter (selection of a specific time interval), spatial filter (definition of a "window" in space), characteristic-based filter (selection of objects, based on their characteristics, such as length or curvature of the objects path) [24]. This allows a preselection of objects.

All three methods can be flexibly adapted and applied in different steps of the KD-process according to the underlying problem formulation and the manifestation of *a priori* knowledge in quantity and quality.

Interactive Modifications

To optimize the allocated information in an iterative process, the possibility of interactive modification is required. If the results of the information allocation process are not satisfactory, the user has the ability to modify the results using his *a priori* knowledge. Mainly, the interactive modifications are used in the context of clustering. Here the user has the possibility to exclude objects belonging to one or several subclusters, fuse clusters or force a further clustering of selected clusters. Further options are to divide subclusters into smaller subclusters or to merge two or more subclusters to one larger subcluster. Moreover, there is the possibility to dissolve one or more subclusters and for example spread their members among the remaining subclusters [26]. The whole modification process has to be supported by appropriate visualization techniques and the possibility for appropriate interactions, e.g. focusing on subclusters or selecting one or more subclusters. Based on this, the user can optimally drive the KD-process to gain knowledge according to the underlying problem formulation.

Application of the Knowledge Discovery Process

After the KD-process is designed it can be applied to new object databases. For a more detailed view of the process, see Figure 3. In each step, objects of the new database are classified according to the designed KD-process. Due to inhomogeneity and differences between the object databases, the user can, if necessary, modify the parameters within each step of the applied KD-process. This allows coping with small differences within various object databases.

To handle this problem, interactive modifications in each step can adapt the classification process. On a very high level of detail, objects from new object databases can often not be classified well, because the designed KD-process is based on another object database. To cope with this problem, the user can interact and add new information allocation steps. The result of applying the designed KD-process to new object databases is a new generated knowledge database.

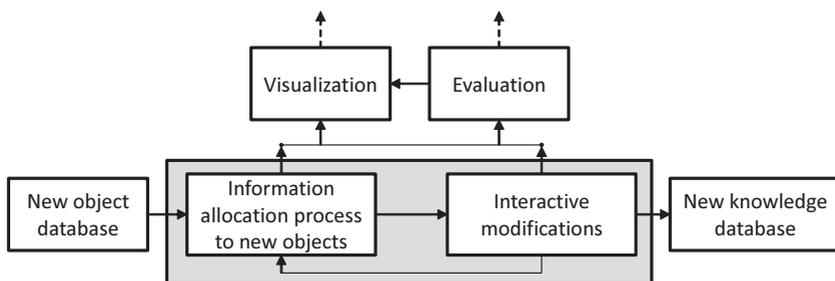


Figure 3: General scheme for the application of a KD-process. Here the process is displayed to create a new knowledge database out of a new generated object database

Comparison of Different Object Databases

For each object database a knowledge database is created using the KD-process. These knowledge databases can be compared and evaluated finally. Here quantitative statements about the comparison of the knowledge databases can be derived. This can serve as a basis for investigating for example different influences to object databases. Such influences are for example the application of different toxic compounds to biological organisms.

Application

In the following, the knowledge discovery framework is exemplarily applied to a benchmark object database to demonstrate its functionality. The overall problem formulation serves as a basis for generating the knowledge database. In some cases this is also possible using a one-step approach, instead of a multi-step framework. However, the approach would be much more complex and not interpretable any more. Further, the *a priori* knowledge within the human brain can hardly be transferred in the way that it can be implemented in such an one-step approach.

In the following sections, the stepwise integration of the *a priori* knowledge in the KD-process is explained in more detail (see Figure 4). Finally, the results of the KD-process and further analysis steps are described. All methods and analysis task used, are implemented using the Matlab toolbox Gait-CAD [3].

Object Database Generation

As an input for the knowledge discovery framework a benchmark object database is generated on the basis of object positions that were automatically extracted from real world biological raw object data [30]. Therefore, object data was extracted from light-sheet fluorescence microscopy images [3, 1]. For the benchmark generation, simulated objects (fluorescently labeled cell nuclei) are placed at randomly selected locations of real objects and temporal dynamics of the objects are simulated. For more details see the description in [31]. The advantage of such a semi-synthetic benchmark is the direct availability of an error-free ground truth combined with realistic object movement behavior. Additionally, the benchmark simulation framework allows the generation of artificially flawed data and different scenarios [31]. In Figure 5, the generated benchmark object database is shown. Here, 520 objects can be observed that exhibit complex movement patterns. Each object has 400 time points and the time series database has a size of approximately 8 MB and the related 3D image database has a size of approximately 250 GB.

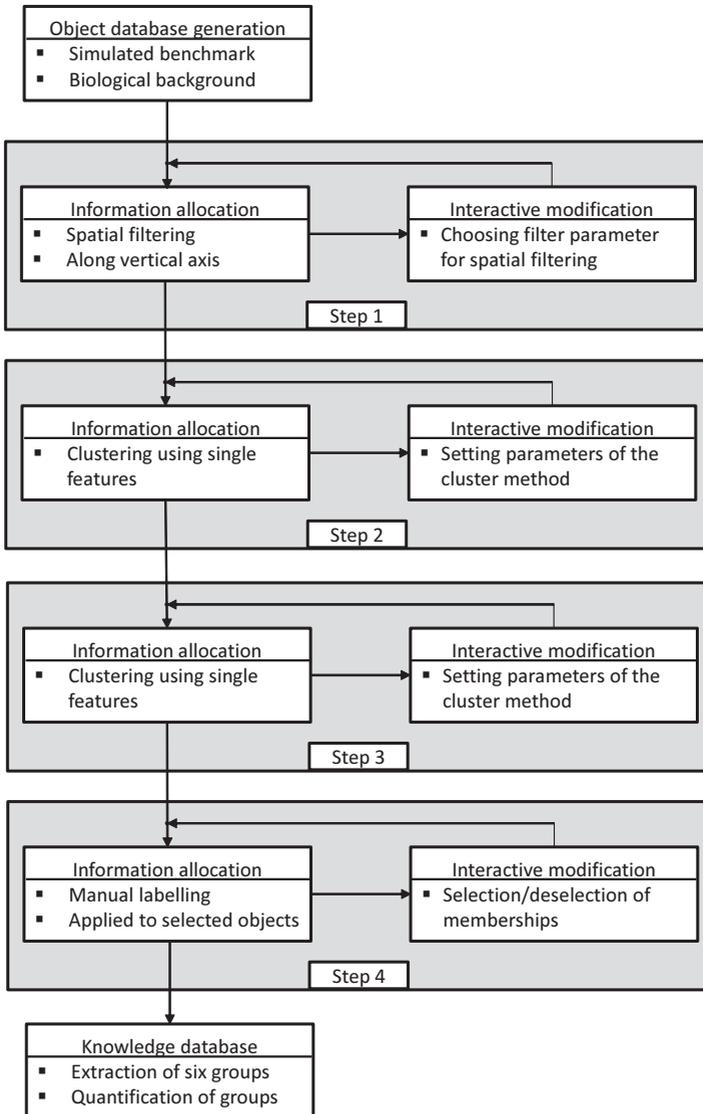


Figure 4: Stepwise application of the KD-process. The "information allocation" and "interactive modification" processes for each step is listed.

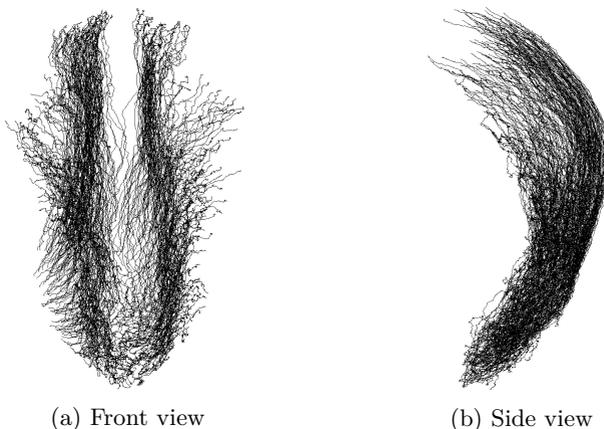


Figure 5: The generated Benchmark database. Here the trajectories of moving objects are displayed. The method from [31] is used.

Overall Problem Formulation

The overall artificial problem formulation to serve as a basis for the knowledge discovery framework is derived from current real world biological issues. The problem formulation comprised the discovery of six different groups of objects according their qualitatively known spatial occurrence and movement direction in a database containing also many other objects. Here three groups on each side (left and right according the vertical axis in Figure 5) of the benchmark have to be extracted. The step-wise problem formulation to extract these groups is described in more detail. Based on the overall problem formulation the aim of the knowledge discovery framework is to extract the corresponding knowledge database.

Design of the Knowledge Discovery Process

The overall problem formulation is integrated step by step within the design of the KD-process to extract the interesting groups of objects. Through this step-wise approach the *a priori* knowledge can be integrated efficiently. In the following the single steps of the iterative design of the KD-process

are described in detail. Thereby, the structure within each step contains the verbal problem formulation and the transformation to the formal one, as well as the description of the information allocation process and the interactive modifications that are applied. Finally at the end of each step, the results are highlighted visually.

Step 1 - Interactive filtering: All the objects that are of interest, are located spatially in the upper half according the vertical axis in Figure 5. This verbal problem formulation is transformed to the formal problem formulation task of spatial filtering using a threshold applied to the values of the y-axis (equal the vertical axis) of the data.

For the information allocation process, interactive filtering with a threshold is used. This threshold can be interactively modified by the user. Through visual feedback of the filter result, the user can modify the threshold in the way, that the visualization of the objects agrees with his *a priori* knowledge. The result of interactive filtering is shown in Figure 6.

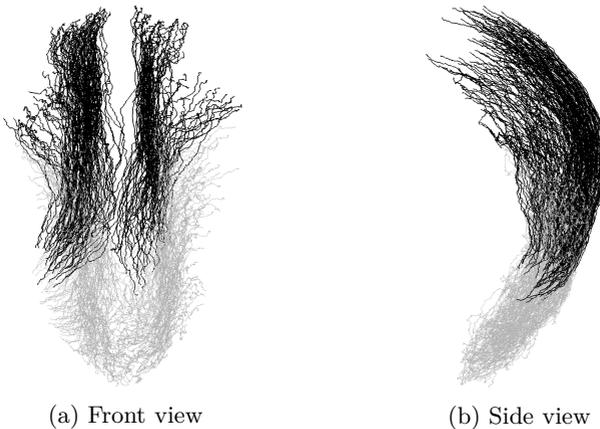


Figure 6: Benchmark database: Interactive spatial filtering according the vertical axes. For further analysis the black printed object trajectories are used.

Step 2 - Separation into two sides: The verbal problem formulation is to separate the objects in two halves (left and right assuming a symmetric object structure) in the horizontal direction. The transformed formal problem

formulation is to extract the start and end point of each object to separate them. Based on these characteristics a fuzzy *c*-means clustering using the start points of the objects as single features (*x*-,*y*- and *z*-component) with two cluster centers is applied to the object data (black highlighted in Figure 6). The settings of the cluster method can be used for further interactive modifications. For further steps within the KD-process the two separated halves are considered as a unit (the information of objects belonging to each half is stored for a later splitting of the extracted groups), however the information of the objects belonging to the both sides is retained. The separation into the both halves is shown in Figure 7.

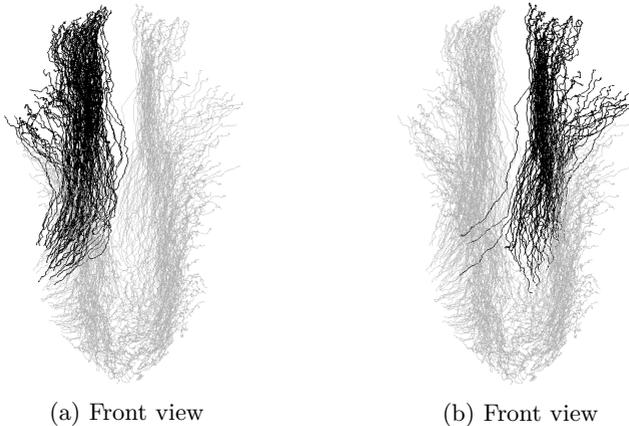


Figure 7: Benchmark database: Separation according to the sides using clustering: (a) left side; (b) right side

Step 3 - Cluster-based separation: Based on the objects of interest (black highlighted objects in Figure 6) the user wants to further distinguish between two groups. Described in a verbal problem formulation, the first group contain objects having a more straight movement in combination with a location near the "surface". The objects not following these characteristics belong to the second group. The verbal task of having "more straight movement" is transformed to the formal problem formulation (the term "problem formulation" deviated from the original definition in [32]) of calculating the angle of each object trajectory to a reference plane (in

this case orthogonal to the front view in Figure 8). Further, the verbal condition of a "location near the surface" is transformed into calculating a single feature that is the range of motion of the objects rotated projection (the whole benchmark is rotated according a predefined angle) on the plane orthogonal to the front view in Figure 8. These two features extracted in the formal problem formulation are further used for clustering. Again a fuzzy *c*-means cluster method is used with two cluster centers (there are two groups to distinguish). Further the cluster parameters can be used for interactive modifications. Figure 8 shows the group having a more straight movement and is located near the surface (right hand side in the side view). The other group is shown in Figure 9.

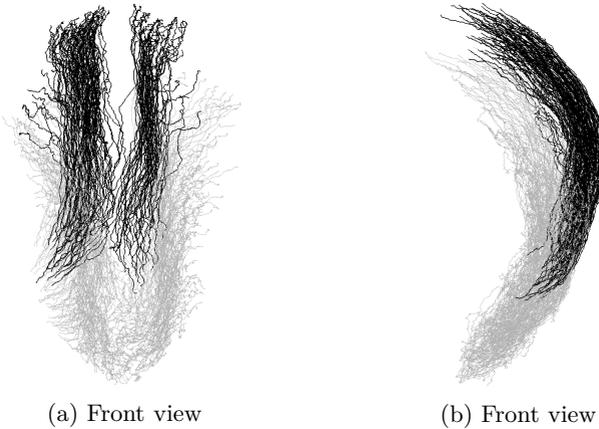


Figure 8: Benchmark database: Separating objects having a straight movement according the vertical axes and are further placed on the "surface" (highlighted in black) against all other objects (colored in grey).

Step 4 - Manual information allocation: The second group, extracted in Step 3 (black highlighted in Figure 9), has to be further separated into two groups. The first group exhibits rather forward directed movements and the second group exhibits rather outward directed movements. Through the complex movement patterns of the objects, the ability is used to allocate the information manually to the objects through labeling. Here, the user's *a priori* knowledge is used directly for information allocation. For an effective

and fast labeling a visual framework was generated supporting the user within the process of information allocation by clicking to each object for a selection or deselection to a group. The results of manual information allocation are displayed in Figure 10.

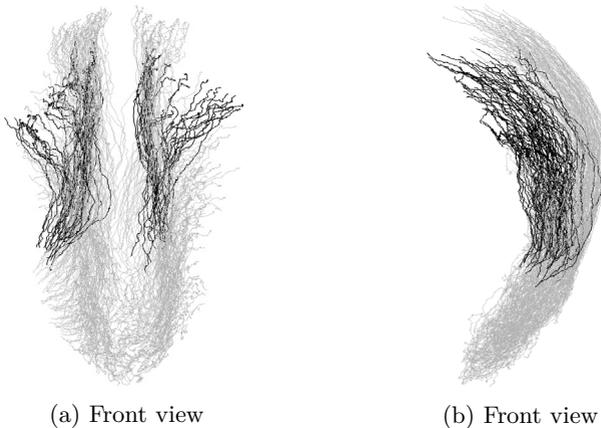


Figure 9: Benchmark database: The objects that do not have a straight movement according the vertical axes and additionally are not placed on the "surface".

Result of the Knowledge Discovery Process

The result of the design of the knowledge discovery framework applied to the benchmark database based on the problem formulation is shown in Figure 12. In this knowledge database the six extracted groups are visualized, each in front and side view. Here Group 1 and 4 can be observed in Figure 10a (left and right side), Group 2 and 3 in Figure 10b (left and right side) and Group 5 and 6 can be observed in Figure 8 (left and right side). Here the heterogeneous spatial distribution of the objects can be shown. Based on the knowledge database further analysis and evaluation steps can be applied. Figure 11 shows one example of the evaluation of the extracted groups according to one chosen characteristic (here the mean curvilinear speed). Similar to this procedure many other characteristics

can be investigated and reveal further insights in the object data. This serve as a basis for application-dependent generation of new knowledge about the behavior and characteristics of objects.

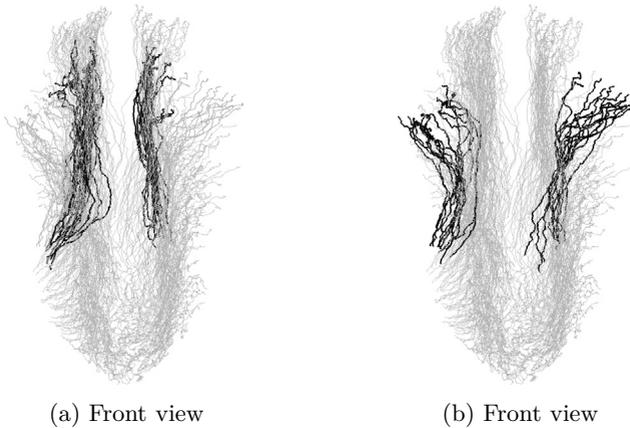


Figure 10: Benchmark database: Result of manual information allocation applied to the black highlighted group in Figure 9. Group of objects having a more straight movement direction (a) and the other group (b).

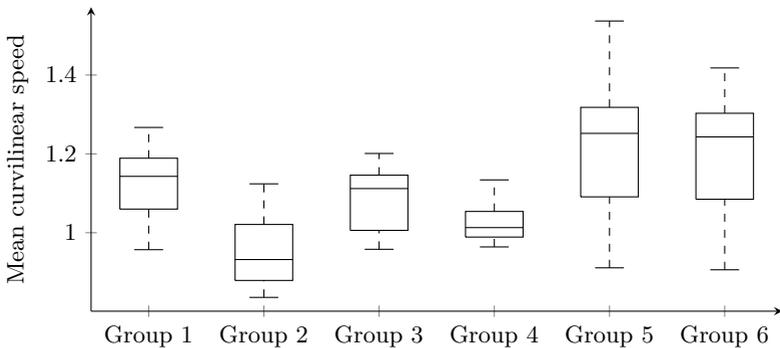


Figure 11: Evaluation of the characteristics "Mean curvilinear speed" (the mean speed from one object position to the following object position) in all groups.

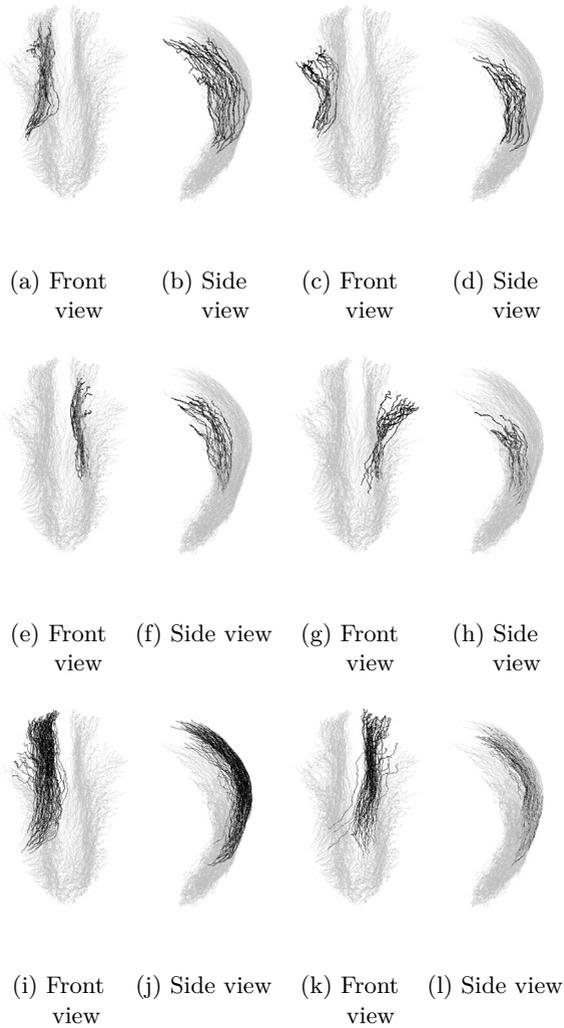


Figure 12: Knowledge database: Extracted groups. Group 1 (a,b), Group 2 (c,d), Group 3 (e,f), Group 4 (g,h), Group 5 (i,j) and Group 6 (k,l).

Conclusion

Using the knowledge discovery framework, it is possible to explore object databases and extract groups of objects with highly heterogeneous movement behavior by efficiently integrating *a priori* knowledge through interacting with the framework. The whole process is modular expandable and is therefore adaptive to any problem formulation. Further, the flexible use of different information allocation processes reveal a great potential to efficiently incorporate the *a priori* knowledge of different users in different ways.

Therefore, the stepwise knowledge discovery process embedded in the knowledge discovery framework is described in detail to point out the flexibility of such a system incorporating object databases from different applications. The described framework can be used to gain knowledge out of object databases in many different fields. This knowledge can be used to gain further insights and improve the understanding of underlying phenomena. The functionality of the proposed framework is exemplarily demonstrated using a benchmark database based on real biological object data.

A further step will be the extension of the modular framework to handle the occurrence of less quantitative and qualitative *a priori* knowledge. Future work will therefore be put on developing new methods for the information allocation process (e.g. create application-dependent standard feature sets or handle incomplete tracks) and to improve the interactive modifications (application-dependent visual frameworks) within the knowledge discovery process. Additionally, the whole framework can be transferred to other real object databases from several fields to extract valuable knowledge out of the underlying objects to get closer insight and increase the understanding of the object database.

References

- [1] Stegmaier, J.; Otte, J. C.; Kobitski, A.; Bartschat, A.; Garcia, A.; Nienhaus, G. U.; Strähle, U.; Mikut, R.: Fast Segmentation of Stained Nuclei in Terabyte-Scale, Time Resolved 3D Microscopy Image Stacks. *PLoS ONE* 9 (2014) 2.

- [2] Andrienko, N.; Andrienko, G.: Spatial Generalization and Aggregation of Massive Movement Data. *IEEE Transactions on Visualization and Computer Graphics* 17 (2011) 2, pp. 205–219.
- [3] Stegmaier, J.; Khan, A.; Reischl, M.; Mikut, R.: Challenges of Uncertainty Propagation in Image Analysis. In: *Proceedings. 22. Workshop Computational Intelligence, Dortmund, 6.-7. Dezember 2012*, pp. 55–70. KIT Scientific Publishing. 2012.
- [4] Beyan, C.; Fisher, R. B.: Detection of Abnormal Fish Trajectories Using a Clustering Based Hierarchical Classifier. In: *British Machine Vision Conference (BMVC), Bristol, UK*. 2013.
- [5] Kobitski, Andrei Y., et al. An Ensemble-Averaged, Cell Density-Based Digital Model of Zebrafish Embryo Development Derived from Light-Sheet Microscopy Data with Single-Cell Resolution. *Scientific reports* 5 (2015).
- [6] Jeung, H.; Yiu, M. L.; Jensen, C. S.: Trajectory Pattern Mining. In: *Computing with spatial trajectories*, pp. 143–177. Springer. 2011.
- [7] Fayyad, U.; Piatetsky-Shapiro, G.; Smith, P.: From Data Mining to Knowledge Discovery in Databases. *AI Magazine* 17 (1996) 3, pp. 37–54.
- [8] Andrienko, G.; Andrienko, N.: Visual Analytics for Geographic Analysis, Exemplified by Different Types of Movement Data. In: *Information Fusion and Geographic Information Systems*, pp. 3–17. Springer. 2009.
- [9] Zhang, Y.; Eick, C. F.: Novel Clustering and Analysis Techniques for Mining Spatio-Temporal Data. In: *Proceedings of the 1st ACM SIGSPATIAL PhD Workshop*, pp. 2. ACM. 2014.
- [10] Li, X.; Hu, W.; Hu, W.: A Coarse-to-Fine Strategy for Vehicle Motion Trajectory Clustering. In: *18th International Conference on Pattern Recognition, 2006. ICPR 2006*, Bd. 1, pp. 591–594. IEEE. 2006.
- [11] Dahlbom, A.; Niklasson, L.: Trajectory Clustering for Coastal Surveillance. In: *10th International Conference on Information Fusion, 2007*, pp. 1–8. IEEE. 2007.

- [12] Gariel, M.; Srivastava, A. N.; Feron, E.: Trajectory Clustering and an Application to Airspace Monitoring. *IEEE Transactions on Intelligent Transportation Systems* 12 (2011) 4, pp. 1511–1524.
- [13] Gudmundsson, J.; Laube, P.; Wollé, T.: Movement Patterns in Spatio-temporal Data. In: *Encyclopedia of GIS*, pp. 726–732. Springer. 2008.
- [14] Fangerau, J.: Interactive Similarity Analysis for 3D+ t Cell Trajectory Data (2015).
- [15] Braga, R. B.; Tahir, A.; Bertolotto, M.; Martin, H.: Clustering User Trajectories to Find Patterns for Social Interaction Applications. In: *Web and Wireless Geographical Information Systems*, pp. 82–97. Springer. 2012.
- [16] Dodge, S.; Weibel, R.; Laube, P.: Exploring Movement-Similarity Analysis of Moving Objects. *SIGSPATIAL Special* 1 (2009) 3, pp. 11–16.
- [17] Atev, S.; Miller, G.; Papanikolopoulos, N. P.: Clustering of Vehicle Trajectories. *IEEE Transactions on Intelligent Transportation Systems* 11 (2010) 3, pp. 647–657.
- [18] Kisilevich, S.; Keim, D.; Rokach, L.: *A Novel Approach to Mining Travel Sequences Using Collections of Geotagged Photos*. pp. 163–182. Springer Berlin Heidelberg. 2010.
- [19] Anjum, N.; Cavallaro, A.: Trajectory Clustering for Scene Context Learning and Outlier Detection. In: *Video Search and Mining*, pp. 33–51. Springer. 2010.
- [20] Liao, T. W.: Clustering of Time Series Data - A Survey. *Pattern recognition* 38 (2005) 11, pp. 1857–1874.
- [21] Junejo, I. N.; Javed, O.; Shah, M.: Multi Feature Path Modeling for Video Surveillance. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*, Bd. 2, pp. 716–719. IEEE. 2004.
- [22] Yanagisawa, Y.; Satoh, T.: Clustering Multidimensional Trajectories Based on Shape and Velocity. In: *22nd International Conference on Data Engineering Workshops, 2006. Proceedings*, pp. 12–12. IEEE. 2006.

- [23] Jiang, F.; Wu, Y.; Katsaggelos, A. K.: Abnormal Event Detection from Surveillance Video by Dynamic Hierarchical Clustering. In: *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, Bd. 5, pp. V–145. IEEE. 2007.
- [24] Rinzivillo, S.; Pedreschi, D.; Nanni, M.; Giannotti, F.; Andrienko, N.; Andrienko, G.: Visually Driven Analysis of Movement Data by Progressive Clustering. *Information Visualization* 7 (2008) 3-4, pp. 225–239.
- [25] Biliotti, D.; Antonini, G.; Thiran, J. P.: Multi-Layer Hierarchical Clustering of Pedestrian Trajectories for Automatic Counting of People in Video Sequences. In: *Seventh IEEE Workshops on Application of Computer Vision, 2005, Volume 1*, Bd. 2, pp. 50–57. IEEE. 2005.
- [26] Andrienko, G.; Andrienko, N.; Rinzivillo, S.; Nanni, M.; Pedreschi, D.; Giannotti, F.: Interactive Visual Clustering of Large Collections of Trajectories. In: *IEEE Symposium on Visual Analytics Science and Technology, 2009*, pp. 3–10. IEEE. 2009.
- [27] Enguehard, R.; Fowler, B.; Hoeber, O.; Devillers, R.; Banzhaf, W.: Integrating Human Knowledge Within a Hybrid Clustering Classification Scheme for Detecting Patterns Within Large Movement Data Sets. *Proceedings of AGILE 2012* (2012).
- [28] Yu, Y.; Wang, Q.; Wang, X.: Continuous Clustering Trajectory Stream of Moving Objects. *China Communications* (2013), pp. 121.
- [29] Piciarelli, C.; Foresti, G. L.: On-line Trajectory Clustering for Anomalous Events Detection. *Pattern Recognition Letters* 27 (2006) 15, pp. 1835–1842.
- [30] Bartschat, A.; Hübner, E.; Reischl, M.; Mikut, R.; Stegmaier, J.: XPIWIT - An XML Pipeline Wrapper for the Insight Toolkit. *Bioinformatics* (2015). In press.
- [31] Stegmaier, J.; Arz, J.; Schott, B.; Otte, J. C.; Kobitski, A.; Nienhaus, G. U.; Strähle, U.; Mikut, R.: Generating Semi-Synthetic Validation Benchmarks for Embryomics. *submitted* (2016).
- [32] Mikut, R.: Data Mining in der Medizin und Medizintechnik. Universitätsverlag Karlsruhe (2008).

Shift-Invariant Feature Extraction for Time-Series Motif Discovery

Sahar Torkamani, Volker Lohweg

inIT – Institute Industrial IT, Ostwestfalen-Lippe University of Applied
Sciences

Langenbruch 6, D-32657 Lemgo

E-Mail: {sahar.torkamani, volker.lohweg}@hs-owl.de

Abstract

The means of data mining and machine learning tasks are important topics in signal processing fundamentals. An example of such tasks is motif discovery. This paper presents an efficient method for shift-invariant feature extraction in time-series motif discovery. The proposed method initiates from the machine learning procedure and tackles the drawbacks of existing methods. Moreover, the efficacy of the novel approach is benchmarked against various algorithms and data from diverse fields.

Introduction

The growth of complex automated production systems has gained huge appeal for machine diagnosis and condition monitoring during recent years. In order to analyse the behaviour of such systems, data mining and machine learning tasks are essential. Motif discovery belongs to the mentioned tasks and its aim is to find meaningful, new, and even unknown knowledge from the data. The term *motif* was first triggered by Patel et al. [1], and means frequently repeated unknown patterns in a signal. Motifs provide valuable insights about the investigated problem to the user.

In the past decade comprehensive research has been performed on this topic [2, 3]. However, in spite of extensive research, only few motif discovery algorithms are able to detect shifted and multi-scale motifs of different size.

In this paper, we propose a method to find shifted and multi-scale patterns or motifs of different size in signals. The core of this method is based on a shift-invariant feature extracting method. As, features present the special characters of the data, therefore it is important that the extracted features from the data are detectable under changes in scale, location or even under noise circumstances.

Wavelet transformations are one of the promising techniques for extracting features in time and frequency domain, and also for noise reduction. However, it is well known that the Discrete Wavelet Transform (DWT) [4] and orthogonal filter banks suffer from a lack of shift invariance, due to the down-sampling operations [4].

To overcome this limitation, we propose a wavelet transformation namely the *Quad Tree-Complex Wavelet Packet* (QT-CWP). It will be shown that the QT-CWP has the shift-invariance property. In addition, multi-resolution, perfect reconstruction, and efficient implementation are other properties of the QT-CWP. The proposed method in this paper benefits from the shift-invariant property of the QT-CWP by extracting features from signals. Here, first the time-series data are sent as input to the QT-CWP, next the proper features are extracted from the wavelet coefficients and finally motifs of different size are identified by measuring the similarity of their features.

This paper is organized as following: First, a review of the existing work and necessary definitions are provided. Next, the proposed approach is explained. Afterwards, the efficiency of this method is evaluated, and finally, the directions of the future work and a conclusion is indicated.

Preliminaries

Related Work

As stated, motif discovery's aim is finding frequent unknown patterns in a signal without any prior information about their locations or shapes. Most of the algorithms for time-series motif discovery face one or more of the following problems: quadratic or exponential time-complexity, definition of static length of motifs, data streaming and finally, multi-scale motifs.

Many researchers aim to handle these problems, for example Mueen [5]

applies the symmetry and triangular inequality properties of the Euclidean distance to improve time complexity. To overcome data streaming, Lin and Li [6] suggest a grammar-based algorithm to find approximate motifs in medical data, by continually generating a grammatical rule for each subsequence. This method can enhance the problem of motif length's definition, but different parameters must be defined for this method to work.

Defining an optimal motif's length is one of the main problems, usually needs expert knowledge. Variable Length Motif Discovery (VLMD), a non-parametric method is presented in [7] that applies a sliding window of different lengths several times to find motifs from shortest to longest length, and later classifies the similar motifs into a group. Unfortunately, this method has quadratic time-complexity.

Many motif discovery algorithms find motifs only at a single resolution. Castro and Azevedo [8] find motifs in different resolutions by a symbolic representation method. Vespier et al. [9] introduced a method to find motifs in multiple temporal scales. However despite the vast research in this area, the existing algorithms suffer of finding multi-scale motifs. Thus, we propose a method to detect shifted and multi-scale motifs of different size in signals by measuring the similarity of their features.

Aforementioned, wavelet transformations are one of the promising techniques for extracting features in time and frequency domain. The DWT is successfully applied in various tasks in image and signal processing during recent decades. However, the DWT's major problem is being shift variant, meaning small shifts in the input signal cause serious distortion between DWT coefficients in different scales. As a result, the DWT is not suitable for feature extraction oriented tasks.

In particular, the issue of shift invariance has been addressed by Kingsbury in [10], where the dual tree-complex wavelet transform (DT-CWT) is presented. Our previous work with the DT-CWT for multi-scale motifs discovery [11] showed that reasonable features can be extracted by the DT-CWT, which are suitable to discover multi-scale motifs. Unfortunately, due to the definition in [12], the captured features are not shift-invariant. Kingsbury [12] defined the shift invariant property of the dual tree-complex wavelet transform by examining the reconstructed wavelet coefficients in every scale. For simplicity, we name this property *shift-tolerance*. In this work, in order to evaluate the shift-invariant property, the wavelet coefficients are examined in original in each scale, and *not* in the reconstructed manner.

Accordingly, it will be proved that the DT-CWT is not shift-invariant. One of the properties of the DT-CWT is analysing a signal by decomposing the low frequency parts in each scale. This frequency decomposition can be a limitation for some applications or signals. One way to overcome this problem is by decomposing both low and high frequency parts in each scale, same as the discrete wavelet packet.

The wavelet packet transform presented in [13], is shift-invariant. However, it cannot provide an analytic representation of a signal. Additionally, the best basis is determined by performing an exhaustive search. Finally, the transform is data-dependent.

In [14], the authors apply the straightforward way to generate the dual tree-wavelet packet by extending the high frequency parts. However, their approach is highly redundant, since the down-samplers are eliminated. Moreover, although this approach is shift-invariant, but the reconstruction is not possible due to filters of different length.

Bayram and Selecnick in [15], introduced a complex version of the wavelet packet, called dual tree-complex wavelet packet (DT-CWP), which is based on the dual-tree approach. However, this wavelet transformation is also not shift-invariant. In order to tackle this limitations, the *Quad Tree-Complex Wavelet Packet* (QT-CWP) is proposed, and explained in the following sections.

Definitions and Notations

Preceding, wavelet transforms provide a flexible time-frequency window, which shrinks in observation of high frequency phenomena and widens in the case of low frequency behaviour. This property makes them suitable to find multi-scale patterns. Above, it is stated that the proposed wavelet transform is based on the structure of the DT-CWT. The structure of the DT-CWT is based on two parallel filterbank trees; namely “tree A” and “tree B”, where, “tree A” represents the real part and “tree B” provides the imaginary part.

Definition 1. *Let $\psi_a(t)$, $\psi_b(t)$ and $\phi_a(t)$, $\phi_b(t)$ be the wavelet and scaling functions of the DT-CWT. For convenience both wavelet transforms are considered as orthonormal. The wavelet and scaling functions in “tree A”,*

$\forall n \in \mathbb{N}$ are given by

$$\psi_a(t) = \sqrt{2} \sum_{n=0}^M {}^s h_a[n] \phi_a(2t - n), \quad \phi_a(t) = \sqrt{2} \sum_{n=0}^M {}^s g_a[n] \phi_a(2t - n).$$

Similarly for “tree B” the wavelet and scaling functions are defined.

$$\psi_b(t) = \sqrt{2} \sum_{n=0}^M {}^s h_b[n] \phi_b(2t - n), \quad \phi_b(t) = \sqrt{2} \sum_{n=0}^M {}^s g_b[n] \phi_b(2t - n),$$

where filters ${}^s h_a$, ${}^s h_b$ are high-pass and ${}^s g_a$, ${}^s g_b$ are low-pass filters, and $s \in \mathbb{N}$ is number of scales. All filters are causal so ${}^s h_{a,b}[n] = 0$ and ${}^s g_{a,b}[n] = 0$ for $n < 0$.

The filters of the DT-CWT form a *Hilbert pair* due to their design [12].

Definition 2. Wavelets ψ_a and ψ_b with the following property

$$\Psi_a(j\omega) = \begin{cases} -j\Psi_b(j\omega), & \omega > 0, \\ j\Psi_b(j\omega), & \omega < 0, \end{cases}$$

are called the Hilbert pair, where $\Psi(j\omega)$ is the Fourier transform of $\psi(t)$.

Theorem 1 (Half-sample delay [16]). Wavelets ψ_a and ψ_b form a Hilbert pair, if the filters ${}^s g_a$ and ${}^s g_b$ satisfy the condition [16],

$${}^s G_a(e^{j\omega}) = {}^s G_b(e^{j\omega})e^{-j\frac{\omega}{2}}. \quad (1)$$

Eq. 1, can be presented in terms of the magnitude and phase functions:

$$|{}^s G_a(e^{j\omega})| = |{}^s G_b(e^{j\omega})|, \quad \angle {}^s G_a(e^{j\omega}) = \angle {}^s G_b(e^{j\omega}) - \frac{1}{2}\omega, \quad (2)$$

which is the so-called “half-sample delay” condition between two low-pass filters ${}^s g_a$, ${}^s g_b$.

Proof. Proof is represented by Selsnick in [16]. □

This means that, the scaling low-pass filters must be offset from one another by a half sample. Moreover, Yu and Ozkaramanli [17] proved that Eq. 2 is the necessary and sufficient condition for two wavelets to form a Hilbert transform pair.

Definition 3 (q-shift filters [18]). *Kingsbury’s solution for design suitable filters for the DT-CWT is called “q-shift”, which satisfy the “half-sample delay” condition given in Theorem 1, where the low-pass filters set as*

$${}^s g_a[n] = {}^s g_b[M - 1 - n]. \quad (3)$$

Here, $M \in \mathbb{N}^+$ is the even length of filter ${}^s g_b$, which is supported on $0 \leq n \leq M - 1$.

As stated in [12], it turns out that the implementation of the DT-CWT needs to use other type of filters than the aforementioned one in the first scale. Since, if the same filters as stated above, are used for each scale, then the first scale will not be analytic [12]. Therefore, the filters ${}^1 g_b$ and ${}^1 g_a$, should have one sample delay difference [12]. This can be satisfied exactly by using the same set of filters in each of the two trees; it is necessary only to translate one set of filters by one sample with respect to the other set.

Definition 4 (First scale DT-CWT’s filter). *The filters for the first scale of the DT-CWT are obtained by*

$${}^1 g_a[n] = {}^1 g_b[n - 1]. \quad (4)$$

Similarly, $M \in \mathbb{N}^+$ is the even length of filter ${}^1 g_b$ which is supported on $\forall n \in \mathbb{N}, 0 \leq n \leq M - 1$. The first scale high-pass filters are given by [12]

$${}^1 h_a[n] = {}^1 h_b[n - 1]. \quad (5)$$

All of these filters are of even length and symmetric. Already stated, the DT-CWT is shift-tolerance, since it analyses the reconstructed wavelet coefficients. Our shift-invariant definition is given by

Definition 5 (shift-invariant). *The shift-invariant is defined by studying the wavelet coefficients of every scale s from both the original and translated signal. This means if $x[n]$ and $x[n - S]$ are respectively the original and translated signal shifted by $S \in \mathbb{Z}$, then the corresponding wavelet coefficients are given by ${}^s d[n]$ and ${}^s d[n, S]$. Let the wavelet transformation be presented by $x[n] \mapsto {}^s d[n]$, then this transformation must satisfy $x[n - S] \mapsto {}^s d[n, S]$, where ${}^s d[n, S] = {}^s d[n - S]$.*

Shift-invariant Property of the DT-CWT

Based on Def. 5, the DT-CWT is only shift invariant in the first scale. In scales greater than one, the wavelet coefficients of the original signal and its shifted ones are not identical to each other. In the following the effect of shifts in the wavelet coefficients of the DT-CWT will be explained in detail. We begin with the wavelet coefficients in the first scale.

Scale=1. The DT-CWT's wavelet coefficients of scale s , ${}^s d[n]$, $n, s \in \mathbb{N}$ are obtained from both "tree A " and "tree B ". Whereby, "tree A " provides the real part and "tree B " the imaginary part of the wavelet coefficients ${}^s d[n] = \text{Re}({}^s d[n]) + j \text{Im}({}^s d[n])$.

The filter bank implementation of the wavelet coefficients are obtained by convolving $x[n]$ with the filter ${}^s h_{a,b}[n]$ and then down-sampling the result by factor 2^s , $\downarrow 2^s$:

$$\begin{aligned} {}^s d[n] &= ((x * {}^s h_a)[n]) \downarrow 2^s + ((x * {}^s h_b)[n]) \downarrow 2^s \\ &= \sum_{k=0}^{L+M-1} x[2n-k] {}^s h_a[k] + j \sum_{k=0}^{L+M-1} x[2n-k] {}^s h_b[k], \quad k \in \mathbb{Z}, \end{aligned} \quad (6)$$

the filters ${}^s h_{a,b}[k] = 0$ for $k < 0$ are high-pass causal filters of "tree A, B ". The wavelet coefficients ${}^1 d[n]$ from "tree A " (real part) are obtained by:

$$\text{Re}({}^1 d[n]) = ((x * {}^1 h_a)[n]) \downarrow 2 = \sum_{k=0}^{L+M-1} x[2n-k] {}^1 h_a[k], \quad (7)$$

and wavelet coefficients of "tree B " (imaginary part) are attained by

$$\text{Im}({}^1 d[n]) = ((x * {}^1 h_b)[n]) \downarrow 2 = \sum_{k=0}^{L+M-1} x[2n-k] {}^1 h_b[k]. \quad (8)$$

According to the definitions, the following corollary presents the effect of shifts in first scale wavelet coefficients of the DT-CWT.

Corollary 2. Assume $x[n]$ is a discrete signal and let $S_{e/o} \in \mathbb{Z}$ be shifts occurred on signal $x[n]$, $S_{e/o}$ can be even, $S_e = 2m$, or odd, $S_o = 2m - 1$, $m \in \mathbb{Z}$. The wavelet coefficients of $x[n - S_e]$ and $x[n - S_o]$ in scale 1, are

depicted by ${}^1d_e[n, S_e]$ and ${}^1d_o[n, S_o]$ respectively and given by:

$$\begin{aligned} {}^1d_e[n, S_e] &= \operatorname{Re}({}^1d_e[n, S_e]) + j \operatorname{Im}({}^1d_e[n, S_e]) \\ &= \sum_{k=0}^{L+M-1} x[2n - k - S_e] {}^1h_a[k] + j \sum_{k=0}^{L+M-1} x[2n - k - S_e] {}^1h_b[k], \\ {}^1d_o[n, S_o] &= \operatorname{Re}({}^1d_o[n, S_o]) + j \operatorname{Im}({}^1d_o[n, S_o]) \\ &= \sum_{k=0}^{L+M-1} x[2n - k - S_o] {}^1h_a[k] + j \sum_{k=0}^{L+M-1} x[2n - k - S_o] {}^1h_b[k], \end{aligned}$$

Then, the following equations hold.

$$\left\{ \begin{array}{l} \forall x[n - S_e], \\ \forall x[n - S_o], \end{array} \right. \left\{ \begin{array}{l} \operatorname{Re}({}^1d[n - \frac{S_e}{2}]) = \operatorname{Re}({}^1d_e[n, S_e]), \\ \operatorname{Im}({}^1d[n - \frac{S_e}{2}]) = \operatorname{Im}({}^1d_e[n, S_e]), \\ \operatorname{Re}({}^1d[n - \lfloor \frac{S_o}{2} \rfloor]) = \operatorname{Im}({}^1d_o[n, S_o]), \\ \operatorname{Im}({}^1d[n - (\lfloor \frac{S_o}{2} \rfloor + 1)]) = \operatorname{Re}({}^1d_o[n, S_o]), \end{array} \right. \quad (9)$$

Proof. Proof is provided in Appendix A.

Scale > 1 . In order to present the effect of shift in wavelet coefficients of scales greater than 1, the following definitions are necessary.

Definition 6. Let K be a set of elements given by $K = \{k_1, k_2, k_3, k_4, \dots\}$. Further, defined $K\{i\} := k_i$.

Axiom 1. Assume R be an equivalence relation, on the set $S_e = \{2, 4, 6, 8, \dots\}$ defined by pRq iff $p \equiv q \pmod{2^s}$, $\forall p, q \in S_e$. Then, R divides the set S_e into $NK = 2^{s-1}$ equivalence classes where each class is presented by ${}^eK_{s,i}$. Parameter s shows the scale under investigation and $0 \leq i \leq NK - 1$ is number of classes. Similarly, R divides the set $S_o = \{1, 3, 5, 7, \dots\}$ into $NK = 2^{s-1}$ equivalence classes, presented by ${}^oK_{s,i}$ $i \in NK$.

For example the number of classes in second scale $s = 2$, for even shifts is equal to two.

$${}^eK_{2,0} = \{0, 4, 8, 12, \dots\}, \quad {}^eK_{2,1} = \{2, 6, 10, 14, \dots\},$$

Corollary 3. Let S_e and S_o be even and odd shifts. In the scales greater than one, the relation between the wavelet coefficients of the original and shifted signals are given as:

- $\forall x[n - S_e]$,

$$\left\{ \begin{array}{l} \mathbf{a.} \forall S_e \in {}^eG_{s,0}, \\ \quad \text{Re}({}^s d[n]) = \text{Re}({}^s d_e[n - \frac{S_e}{2^s}]), \\ \mathbf{b.} \forall S_e \in {}^eG_{s,i} \ i \neq 0, i \leq NK - 1, \\ \quad \text{Re}({}^s d[n, {}^eG_{s,i}\{1\}]) = \text{Re}({}^s d_e[n - \lfloor \frac{S_e}{2^s} \rfloor]). \end{array} \right. \quad (10)$$

- $\forall x[n - S_o]$,

$$\mathbf{a.} \forall S_o \in {}^oG_{s,i} \ i \leq NK - 1, \\ \text{Re}({}^s d[n, {}^oG_{s,i}\{1\}]) = \text{Re}({}^s d_o[n - \lfloor \frac{S_o}{2^s} \rfloor]). \quad (11)$$

Proof. Proof is given in Appendix B. □

As proved, the DT-CWT's wavelet coefficients are able to detect the odd and even shift only in the first scale. In rest of the scales, the wavelet coefficients of the original signal and its shifted ones are not identical to each other. However, it is possible to apply the equivalent classes and relations to derive information from the wavelet coefficients, but one need to know that how many shifts are occurred on the original signal, in order to get these information. The upcoming sections provide comprehensive information on the proposed approach.

Proposed Approach

As acknowledged, the motif discovery algorithms suffers from two drawbacks which the proposed method, *Shift-Invariant Feature Extraction for Motif Discovery (SIMD)*, tries to tackle them. Our previous approach governs the DT-CWT in order to handle multiple invariances. Although, reasonable features were extracted by the DT-CWT, for multi-scale motifs discovery. Unfortunately, the captured features were not shift-invariant. As a matter of fact, a wavelet transform is proposed which tackles the drawback of the DT-CWT. This wavelet transform is called the *Quad Tree-Complex Wavelet Packet (QT-CWP)*, and is the extended version of the DT-CWT.

The upcoming section explains the design and properties of this wavelet transforms.

The SIMD adopts a brute force algorithm where all possible pairs of subsequences of all available lengths are obtained by sliding windows of various sizes. Therefore, there is no need for the motif's length definition in advance. Next, these subsequences are sent as inputs to the QT-CWP which provides a comprehensive analysis of a signal into different frequency scales. Afterwards, features are extracted from the wavelet coefficients. Since the core of the suggested approach is based on the pattern recognition, the extracted features from the data should be detectable under changes in scale, location or other distortions. Finally, the derived features are compared to each other using popular distance similarity measures in time-series motif discovery.

The Quad Tree-Complex Wavelet Packet

The QT-CWP consists of two wavelet packet trees working parallel to each other; namely “*WPT A*” and “*WPT B*”. The structure of “*WPT A*” is presented in Fig. 1, where $\downarrow 2^e$ and $\downarrow 2^o$ depict the even and odd down-sampling.

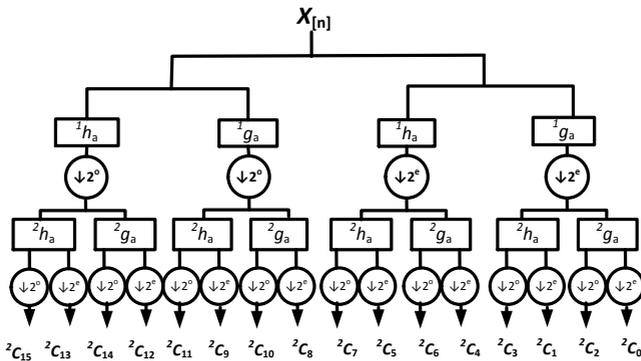


Figure 1: First wavelet packet of a two scale QT-CWP. The second wavelet packet is obtained by replacing the filters $^s g_a, ^s h_a$ with $^s g_b, ^s h_b$

To provide a shift-invariance property, a non shifted and a shifted version of the input signal in each scale are decomposed. This is based on the fact that an arbitrary signal translation of S samples is bounded by $\text{mod}(S, 2)$, because of the down-sampling in each scale.

As stated in [12] an analytic representation of a signal is achieved if the filters form a Hilbert pair. Since, the same filters as in the DT-CWT are employed in the QT-CWP, therefore, the response of each branch of the “WPT A” and the corresponding branch of the “WPT B” forms a Hilbert pair. Consequently, the QT-CWP is analytic in each sub-band. Thus, the structure provides an analytic presentation of the input signal.

The wavelet and scaling coefficients of the QT-CWP for the “WPT A” are defined in Def. 7.

Definition 7. *Coefficients of the QT-CWP for the “WPT A” are given by ${}^s C[n] = \{ {}^{s+1}C_{2J}[n], {}^{s+1}C_{2J+1}[n], {}^{s+1}C_{2J+2}[n], {}^{s+1}C_{2J+3}[n] \}$ and obtained by*

$$\begin{aligned}
 {}^{s+1}C_{2J}[n] &= \sum_{k=0}^{M+Len-1} {}^s g_a[k] {}^s C_j[2n - k], \\
 {}^{s+1}C_{2J+1}[n] &= \sum_{k=0}^{M+Len-1} {}^s h_a[k] {}^s C_j[2n - k], \\
 {}^{s+1}C_{2J+2}[n] &= \sum_{k=0}^{M+Len-1} {}^s g_a[k] {}^s C_j[2n + 1 - k], \\
 {}^{s+1}C_{2J+3}[n] &= \sum_{k=0}^{M+Len-1} {}^s h_a[k] {}^s C_j[2n + 1 - k].
 \end{aligned} \tag{12}$$

where $\text{length}({}^s C_j) = Len$, $J = 2j$, and $0 \leq j < 2^s \cdot (s - 1)$.

The coefficients of the second wavelet packet “WPT B” are given by ${}^s D[n] = \{ {}^{s+1}D_{2J}[n], {}^{s+1}D_{2J+1}[n], {}^{s+1}D_{2J+2}[n], {}^{s+1}D_{2J+3}[n] \}$ and determined the same as in “WPT A”. Nevertheless, the filters ${}^s g_a$ and ${}^s h_a$ must be replaced by ${}^s g_b$ and ${}^s h_b$.

The QT-CWP is shift-invariant, which means the wavelet coefficients of the original signal and its shifted ones are identical to each other. The shift invariance property is proved by the following corollary.

Corollary 4. Assume $x[n]$ is a discrete signal and let $S_{e/o} \in \mathbb{Z}$ be shifts occurred on signal $x[n]$, where $S_{e/o}$ can be even or odd. The QT-CWP wavelet coefficients of $x[n - S_e]$ and $x[n - S_o]$ from “WPT A” in scales s , are depicted by ${}^s C'_e[n, S_e]$ and ${}^s C'_o[n, S_o]$, given by:

$${}^s C'_{e/o}[n, S_{e/o}] = \{ {}^{s+1} C'_{2J}[n, S_{e/o}], {}^{s+1} C'_{2J+1}[n, S_{e/o}], {}^{s+1} C'_{2J+2}[n, S_{e/o}], {}^{s+1} C'_{2J+3}[n, S_{e/o}] \},$$

and for “WPT B” are provided by

$${}^s D'_{e/o}[n, S_{e/o}] = \{ {}^{s+1} D'_{2J}[n, S_{e/o}], {}^{s+1} D'_{2J+1}[n, S_{e/o}], {}^{s+1} D'_{2J+2}[n, S_{e/o}], {}^{s+1} D'_{2J+3}[n, S_{e/o}] \},$$

Then, the following equations hold

$$\left\{ \begin{array}{l} \forall x[n - S_e], \quad \left\{ \begin{array}{l} {}^s C[n] = {}^s C'_e[n - \lfloor \frac{S_e}{2^s} \rfloor], \\ {}^s D[n] = {}^s D'_e[n - \lfloor \frac{S_e}{2^s} \rfloor]. \end{array} \right. \\ \forall x[n - S_o], \quad \left\{ \begin{array}{l} {}^s C[n] = {}^s C'_o[n - \lfloor \frac{S_o}{2^s} \rfloor], \\ {}^s D[n] = {}^s D'_o[n - \lfloor \frac{S_o}{2^s} \rfloor]. \end{array} \right. \end{array} \right. \quad (13)$$

Proof. The coefficients of signal $x[n - S_{e/o}]$ for both odd and even shifts are given in following:

1. Even Shifts. If $x[n - S_e]$ where the shift $S_e = 2m$, $m \in \mathbb{Z}$ then QT-CWP's coefficients ${}^{s+1} C'_{2J}[n, S_e]$ and ${}^{s+1} C'_{2J+1}[n, S_e]$ are able to detect the shift. Thus,

$$\begin{aligned} & {}^{s+1} C'_{2J}[n, S_e] \stackrel{S_e=2m}{=} \sum_{k=0}^{M+Len-1} {}^s g_a[k] {}^s C'_j[2n - 2m - k] = \dots \\ & \dots = \sum_{k=0}^{M+Len-1} {}^s g_a[k] {}^s C'_j[2(n - m) - k] = {}^{s+1} C_{2J}[n - m], \\ & {}^{s+1} C'_{2J+1}[n, S_e] \stackrel{S_e=2m}{=} \sum_{k=0}^{M+Len-1} {}^s h_a[k] {}^s C'_j[2n - 2m - k] = \dots \\ & \dots = \sum_{k=0}^{M+Len-1} {}^s h_a[k] {}^s C'_j[2(n - m) - k] = {}^{s+1} C_{2J+1}[n - m]. \end{aligned} \quad (14)$$

2. Odd Shifts. If $x[n - S_o]$ where the shift $S_o = 2m + 1$, $m \in \mathbb{Z}$ then QT-CWP's coefficients ${}^{s+1}C'_{2J+2}[n, S_o]$ and ${}^{s+1}C'_{2J+3}[n, S_o]$ are able to detect the shift. Thus,

$$\begin{aligned}
& {}^{s+1}C'_{2J+2}[n, S_o] \stackrel{S_o=2m+1}{=} \sum_{k=0}^{M+Len-1} {}^s g_a[k] {}^s C'_j[2n + 1 - 2m - 1 - k] = \dots \\
& \dots = \sum_{k=0}^{M+Len-1} {}^s g_a[k] {}^s C'_j[2(n - m) - k] = {}^{s+1}C_{2J}[n - m], \\
& {}^{s+1}C'_{2J+3}[n, S_o] \stackrel{S_o=2m+1}{=} \sum_{k=0}^{M+Len-1} {}^s h_a[k] {}^s C'_j[2n + 1 - 2m - 1 - k] = \dots \\
& \dots = \sum_{k=0}^{M+Len-1} {}^s h_a[k] {}^s C'_j[2(n - m) - k] = {}^{s+1}C_{2J+1}[n - m].
\end{aligned} \tag{15}$$

Similarly, the coefficients for the second wavelet packet “WPT B” are obtained. \square

Motif Discovery

Motif discovery algorithms are mostly search methods for detecting unknown patterns within time-series. The main task is to define the distinctive features to distinguish classes from each other. The next typical step is to determine the similarity between time-series employing the features.

Feature Extraction

The first four statistical moments namely; mean value, variance, skewness and kurtosis are considered as features. These features are extracted from both the real and imaginary wavelet coefficients. Since, the QT-CWP is shift-invariant then these features have identical values even if the data is shifted. Moreover, applying such features are proved to be useful in most of the signal and image processing applications [19, 20].

Additionally, energy of the wavelet coefficients is considered as another feature. Since based on the Parseval's theorem the energy of the wavelet

coefficients and their shifted ones are similar. Furthermore, the phase of the coefficients $\theta = \arctan\left(\frac{sD[n]}{sC[n]}\right)$ is studied as another features. This feature provides the phase-spectrum characteristic of the original signal. This information makes the complex wavelet transformation superior to the real wavelets.

The efficiency of the extracted features are tested by the linear discriminant linear (LDA) algorithm [21].

Similarity Measures

Most of the motif discovery algorithms use distance similarity measures [2]. Such common methods are : Euclidean distance [22], Dynamic Time Warping [22], and Canberra distance [22]. Euclidean and Canberra distance have linear computational time, and are metric. So, the sub-additivity property speeds up the search of similar time-series motifs in the data set. Additionally, the Euclidean distance is surprisingly competitive with other more complex approaches, especially regarding large data [23]. On the other hand, this measure suffers of being sensible to outliers and misalignments. Dynamic Time Warping matches various sections of a time-series by warping of the time axis. The proper alignment is obtained by the shortest warping path in a distance matrix. Time-complexity of this method is $\mathcal{O}(n^2)$, although several measurements are introduced in literature to advance the computation.

Experimental Evaluation

The practical evaluation is done for both motifs of similar and variable lengths. In order to analyse the obtained results several quality measures are used during the experiments.

The usefulness of the proposed method is demonstrated with data sets, from various domains and with different properties such as amount of noise, length of the time-series, length of the motif.

Moreover, four common used motif discovery algorithms namely Mr. Motif [8], Enumeration motif [5], Mueen-Keogh (MK) [24], and the grammar-based method [6] are applied as benchmarks to evaluate the recent approach.

Quality Measures. A motif is perfect when it matches all the subsequences in the target class/group and no other subsequences out of that class. Various quality measures are applied to examine the result of the proposed motif discovery algorithm. Four possible cases to qualify a motif m matching a subsequent S_p are: True Positive rate (TP), False Negative rate (FN), True Negative rate (TN), and False Positive rate (FP), which can be obtained for each class label. Moreover, other quality measures which can obtain from the four mentioned measures are Sensitivity ($Sn = \frac{TP}{TP+FN}$) which measures the capacity of subsequences of the target class correctly matched by the motif, and Specificity ($Sp = \frac{TN}{TN+FP}$) that calculates the proportion of subsequences outside the target class that are not matched by the motif. Precision ($Pr = \frac{TP}{TP+FP}$) provides the fraction of subsequences of the target class that are matched by the motif and the subsequence that are not correctly matched by the motif. F-Measure ($F - M = 2 \cdot (\frac{Pr \cdot Sn}{Pr+Sn})$) considers both precision and sensitivity. Additionally, we obtain the correct motif discovery rate $CR = \frac{n^+}{N}$ where n^+ is number of correctly detected motifs and N is number of all motifs.

Test Cases. The Gun data set [25] is gathered from the video surveillance domain. Two types of motifs are included in the data set: Gun draw and gun point. For each motif, 100 examples are considered.

Swedish leaf data set [25] comes from a project at Linköping University. The data set contains of leaves from 15 different Swedish trees. In our experiments, only 3 types of leaves are considered as motifs.

AutoSense data is gathered from the running research project called “Adaptive energy self-sufficient sensor network for monitoring safety-critical self-service-systems” [26]. The focus of this project is monitoring security critical systems, e. g., the identification of criminal attacks on automated teller machines (ATMs). After identification of several relevant attacks on ATMs, these attacks are tested and the results of them are gathered from sensors connected to the system. This data consists of 24 signals with different lengths, gathered from 8 sensors in 3 different experiments done on an ATM machine.

Results. Based on the mentioned quality measures, the obtained result of the proposed method and the other benchmark methods for motifs of equal length are given in Tab. 1-3.

The MK algorithm [24] is able to detect only one pair of motif of similar length. The MK algorithm is useful in the case of one motif/pattern data. However, tested data sets here contain motifs/patterns in more than one

classes. The grammar-based method is also able to identify most of the motifs of equal lengths, although one need to match the best parameters combination. For this reason, only the results of Mr. Motif and Enum. Motif are comparable with our results.

As shown, the SIMD method provides better result compared with other algorithms. In all tables the specificity shows high percentage due to the large amount of TN. Also, the percentage of precision in the SIMD is higher than other methods, which depicts the larger amount of correctly found motifs. The F-measure is considered as the overall accuracy measurement which is large for our proposed method in comparison with others.

Table 1: Evaluation results of equal length motifs in *Gun Data* [25], Sn: Sensitivity, Sp: Specificity, Pr: Precision, F-M: F-Measure, CR: Correct motif discovery rate

Method	Sn(%)	Sp (%)	Pr (%)	F-M (%)	CR (%)
SIMD	90.0	90.0	90.0	90.0	90.0
Mr. Motif	70.0	70.0	70.0	70.0	75.0
Enum. Motif	42.5	65.0	54.8	47.9	50.0

Table 2: Evaluation results of equal length motifs in *Swedish Leaf* [25], Sn: Sensitivity, Sp: Specificity, Pr: Precision, F-M: F-Measure, CR: Correct motif discovery rate

Method	Sn(%)	Sp (%)	Pr (%)	F-M (%)	CR (%)
SIMD	94.9	96.8	93.3	94.1	95.0
Mr. Motif	54.2	95.8	86.4	66.6	73.3
Enum. Motif	55.9	85.9	66.0	55.9	55.0

Table 3: Evaluation results of equal length motifs in *AutoSense* [26], Sn: Sensitivity, Sp: Specificity, Pr: Precision, F-M: F-Measure, CR: Correct motif discovery rate

Method	Sn(%)	Sp (%)	Pr (%)	F-M (%)	CR (%)
SIMD	80.8	97.4	81.9	81.4	80.0
Mr. Motif	25.0	95.2	42.8	31.5	54.1
Enum. Motif	62.5	95.7	65.2	63.8	62.5

Table 4: Evaluation results of the SIMD method for variable length motifs, Sn: Sensitivity, Sp: Specificity, Pr: Precision, F-M: F-Measure, CR: Correct motif discovery rate

Data	Sn (%)	Sp (%)	Pr (%)	F-M (%)	CR (%)
Gun	72.5	72.5	72.5	72.5	75.0
Swedish Leaf	94.9	95.8	91.8	93.3	93.3
AutoSense	83.3	97.6	83.3	83.3	73.0

In most of the cases, Mr. Motif algorithm also provides reasonable results. Although, in the case of the AutoSense data the result are not acceptable. This is due to the representation method applied in Mr. Motif which is unable to handle noise in the data.

Additionally, in the second experiments the aforementioned methods are tested to find motifs of variable lengths. However, only the SIMD is able to discover such motifs. Correspondingly, we are not able to compare our method with other algorithms in the case of motifs with multi-variable lengths. Nevertheless, the result of the proposed method for motifs of variable lengths are given in Tab. 4. As depicted, only the proposed method detects motifs of variable length, which proves the success of our algorithm. It should be mentioned that these methods were examined against more data sets in [25], but due to lack of space the results of three data sets are given in this paper.

Conclusion and Outlook

A novel approach for multi-scale time-series motif discovery is proposed. The underlying approach employs a shift-invariant feature extraction method as well as a time-frequency scale-invariant transformation which is able to handle multiple distortion or variances in time-series. The Quad Tree-Complex Wavelet Packet (QT-CWP) is proposed, which overcomes the limitation of the DT-CWT. It is shown that the DT-CWT is not shift-invariant, which means the wavelet coefficients of the signal and its shifted versions are not identical to each other. The QT-CWP decomposes both low and high frequency parts, providing the better information insight.

After decomposing the time-series data, features are extracted from the wavelet coefficients. Finally, motifs are found by comparing the extracted features to each other using different distance similarity measures.

The practical results showed that the proposed approach is able to find squeezed/stretched motifs of variable length. The comparison of our results with most common algorithms proved that this novel approach tackles the drawbacks of existing methods.

The future tasks are considered as following: Since the QT-CWP analyses the signal into different scales and frequency parts, applying a best branch algorithm is essential to find the best scale with the most information content. Furthermore, studying other types of similarity measures is one of the next steps. Finally, semantics and meanings of the found motifs should be investigated in order to detect the most proper motifs and apply them in broader tasks such as classification or anomaly detection.

Appendix A

The first scale wavelet coefficients of signal $x[n - S_o]$ for odd shifts are given in following:

Odd Shifts. If $x[n - S_o]$ where the shift $S_o = 2m + 1$ then the real wavelet coefficients are computed

$$\begin{aligned} \text{Re}({}^1d_o[n, S_o]) &= \sum_{k=0}^{L+M-1} x[2n - k - S_o] {}^1h_a[k] = \dots \\ \dots &= \sum_{k=0}^{L+M-1} x[2n - k - 2m - 1 + (1 - 1)] {}^1h_a[k] \stackrel{{}^1h_a[k] = {}^1h_b[\underline{k-1}], p=k-1}{=} \dots \\ \dots &= \sum_{p=-1}^{L+M-2} x[2(n - m - 1) - p] {}^1h_b[p], \end{aligned}$$

since the high-pass filter ${}^1h_b[p] = 0$ for $p < 0$ and also based on the convolution property (${}^1d_o[n, S_o] = 0$ when $p > M + L - 1$), then

$$\dots = \sum_{p=0}^{L+M-1} x[2(n - m - 1) - p] {}^1h_b[p] = \text{Im}({}^1d[n - m - 1])$$

The imaginary coefficients are obtained by

$$\begin{aligned}
 \text{Im}({}^1d_o[n, S_o]) &= \sum_{k=0}^{L+M-1} x[2n - k - S_o] {}^1h_b[k] = \dots \\
 \dots &= \sum_{k=0}^{L+M-1} x[2(n - m) - k - 1] {}^1h_b[k] \stackrel{p=k+1, {}^1h_a[k]={}^1h_b[k-1]}{=} \dots \\
 \dots &= \sum_{p=0}^{L+M-1} x[2(n - m) - p] {}^1h_a[p] = \text{Re}({}^1d[n - m]).
 \end{aligned}$$

From above equations, it can be concluded that,

$$\text{Re}({}^1d_o[n, S_o]) = \text{Im}({}^1d[n - m + 1]) = \text{Im}({}^1d[n - (\lfloor \frac{S_o}{2} \rfloor + 1)]),$$

$$\text{Im}({}^1d_o[n, S_o]) = \text{Re}({}^1d[n - m]) = \text{Re}({}^1d[n - (\lfloor \frac{S_o}{2} \rfloor)]).$$

Even Shifts. Wavelet coefficients for even shifts are computed similarly.

Appendix B

The relation between the DT-CWT's coefficients in higher scale is given in Cor. 3 and proved here.

Let $s > 1$ and $s = I$, the even numbers are classified into $NK = 2^{I-1}$ classes; such as

$${}^eK_{I,0} = \{0, 0+2^I, \dots\}, {}^eK_{I,1} = \{2, 2+2^I, \dots\}, \dots, {}^eK_{I,NK-1} = \{2^I-2, \dots\},$$

and similarly for the odd numbers. Then for

Even Shifts. As shown S_e can be divided into the NK equivalence classes, given above. Thus,

1. Assume $\forall S_e \in {}^e K_{I,0}$, where $S_e = 2m$ and $m = \{m|\exists\beta, m = 2^{I-1}\beta, \beta \in \mathbb{Z}\}$, then Eq. 7 can be written as

$$\begin{aligned} \operatorname{Re}({}^I d_e[n, S_e]) &= \sum_{k=0}^{L+M-1} x[2^I n - k - 2^I \beta] {}^2 h_a[k] = \dots \\ \dots &= \sum_{k=0}^{L+M-1} x[2^I(n - \beta) - k] {}^2 h_a[k] = \operatorname{Re}({}^2 d[n - \beta]). \end{aligned}$$

2. Assume $\forall S_e \in {}^e K_{I,1}$ where $S_e = 2m$ and $m = \{m|\exists\beta, m = 2^{I-1}\beta + 1, \beta \in \mathbb{Z}\}$. Consequently

$$\begin{aligned} \operatorname{Re}({}^I d_e[n, S_e]) &= \sum_{k=0}^{L+M-1} x[2^I n - k - 2^I \beta - 2] {}^2 h_a[k] = \dots \\ \dots &= \sum_{k=0}^{L+M-1} x[2^I(n - \beta) - k - 2] {}^2 h_a[k] = \operatorname{Re}({}^I d_e[(n - \beta), 2]). \end{aligned}$$

3. Similarly assume $\forall S_e \in {}^e K_{I,NK}$ where $S_e = 2m$ and $m = \{m|\exists\beta, m = 2^{I-1}\beta + I, \beta \in \mathbb{Z}\}$. Consequently

$$\begin{aligned} \operatorname{Re}({}^I d_e[n, S_e]) &= \sum_{k=0}^{L+M-1} x[2^I n - k - 2^I \beta - 2^I + 2] {}^2 h_a[k] = \dots \\ \dots &= \sum_{k=0}^{L+M-1} x[2^I(n - \beta) - k - 2^I + 2] {}^2 h_a[k] = \dots \\ \dots &= \operatorname{Re}({}^I d_e[(n - \beta), 2^I - 2]). \end{aligned}$$

Odd Shifts. The same results can be obtained for the odd shift.

References

- [1] P. Patel, E. Keogh, J. Lin, S. Lonardi, "Mining motifs in massive time series databases". In: *Proceedings IEEE International Conference on Data Mining*. P. 370-377. 2002.
- [2] P. Esling, C. Agon, "Time-series data mining". *ACM Computing Surveys* Vol. 45. No. 1. P. 1-34. 2012.
- [3] T. Fu, "A review on time series data mining". In: *Engineering Applications of Artificial Intelligence*. Vol. 24. No. 1. Elsevier. 2011.
- [4] Y. Meyer, R.D. Ryan, "Wavelets: Algorithms and Applications". Society for Industrial and Applied Mathematics. 1993.
- [5] A. Mueen, "Enumeration of time series motifs of all length". In: *IEEE 13th International Conference on Data Mining*. P. 547-556. 2013.
- [6] J. Lin, Y. Li, "Finding approximate frequent patterns in streaming medical data" In: *IEEE 23rd International Symposium on Computer-Based Medical Systems* P. 13-18. 2010.
- [7] P. Nunthanid, V. Niennattrakul, C.A. Ratanamahatana, "Parameter-free motif discovery for time series data". In: *9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology* P. 1-4. 2012.
- [8] N. Castro, P. J. Azevedo, "Multiresolution Motif Discovery in Time Series". In: *Proceedings of the International Conference on Data Mining* P. 665-676. 2010.
- [9] U. Vespier, S. Nijssen, A. Knobbe, "Mining Characteristic Multi-scale Motifs in Sensor-based Time Series". In: *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management* P. 2393-2398. ACM. 2013.
- [10] N. Kingsbury, "Complex wavelets and shift invariance". In: *IEEE Seminar on Time-scale and Time-Frequency Analysis and Applications*. P. 1-5. 2000.

- [11] S. Torkamani, V. Lohweg, “Identification of Multi-Scale Motifs”. In: *Proceedings. 24. Workshop Computational Intelligence, Dortmund* Vol. 50. KIT Scientific Publishing, 2014.
- [12] I.W. Selesnick, R.G. Baraniuk, N.C. Kingsbury, “The dual-tree complex wavelet transform”. In: *IEEE Signal Processing Magazine* Vol. 22. No. 6. P. 123-151. 2005.
- [13] I. Cohen, S. Raz, D. Malah, “Orthonormal shift-invariant wavelet packet decomposition and representation”. In: *Signal Processing* Vol. 57. No. 3. P. 251-270. Elsevier. 1997.
- [14] T. A. Jilani, S. A. Burney, C. Ardil, “Satellite image deconvolution using complex wavelet packets”. In: *International Journal of Computational Intelligence* Vol. 22. No. 6. P. 123-151. 2005.
- [15] I. Bayram, I. W. Selesnick, “On the Dual-Tree Complex Wavelet Packet and-Band Transforms”. In: *IEEE Transactions on Signal Processing* Vol. 56. No. 6. P. 2298-2310. 2008.
- [16] I.W. Selesnick, “Hilbert transform pairs of wavelet bases”. In: *IEEE Signal Processing Letters* Vol. 8. No. 6. P. 170-173. 2001.
- [17] R. Yu, H. Ozkaramanli, “Hilbert transform pairs of orthogonal wavelet bases: Necessary and sufficient conditions”. In: *IEEE Transactions on Signal Processing* Vol. 53. No. 12. P. 4723-4725. 2005.
- [18] N. Kingsbury, “Design of q-shift complex wavelets for image processing using frequency domain energy minimization”. In: *Proceedings International Conference on Image Processing* Vol. 1. 2003.
- [19] V. Lohweg, J. L. Hoffmann, H. Dörksen, R. Hildebrand, E. Gillich, J. Hofmann, J. Schaede, “Banknote authentication with mobile devices”. In: *IS&T/SPIE Electronic Imaging* International Society for Optics and Photonics. San Francisco, USA. 2013.
- [20] J.R. Delgado-Contreras, J.P. Garcia-Vazquez, R.F. Brena, “Classification of environmental audio signals using statistical time and frequency features”. In: *International Conference on Electronics, Communications and Computers* P. 212-216. 2014.
- [21] W. Kohn, “Statistik: Datenanalyse und Wahrscheinlichkeitsrechnung Statistik und ihre Anwendungen”, Berlin: Springer, 2005.

- [22] M. M. Deza, E. Deza, “Encyclopedia of distances”. Springer. 2009.
- [23] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, E. Keogh, “Querying and mining of time series data: experimental comparison of representations and distance measures” In: *Proceedings of the VLDB Endowment*, Vol. 1. No. 2. P. 1542-1552. 2008.
- [24] A. Mueen, E. Keogh, Q. Zhu, S. Cash, M. B. Westover, “Exact Discovery of Time Series Motifs” In: *Proceedings of the SIAM International Conference on Data Mining* P. 473-484. 2009.
- [25] “The UCR time series classification archive”. Available at: http://www.cs.ucr.edu/~eamonn/time_series_data 2015.
- [26] A. Dicks, V. Lohweg, H. Wittke, S. Linke, “Structural Health Monitoring of Plastic Components with Piezoelectric Sensors”. In: *Proceeding of 20th IEEE International Conference on Emerging Technologies and Factory Automation* Luxembourg. 2015.

Asynchrone Parallelisierung des SMS-EMOA zur Parameteroptimierung von mobilen Robotern

Dino A. Menges¹, Simon Wessing², Günter Rudolph²

¹Adept Technology GmbH
Revierstr. 5, 44379 Dortmund
E-Mail: dino.menges@adept.com

²Fakultät für Informatik, Lehrstuhl für Algorithm Engineering
Technische Universität Dortmund, Otto-Hahn-Str. 14, 44221 Dortmund
E-Mail: {simon.wessing, guenter.rudolph}@tu-dortmund.de

Einleitung

Diese Arbeit behandelt die mehrkriterielle Optimierung von Problemen mit stark variierender Auswertungsdauer am Beispiel der Parameteroptimierung des mobilen Roboters *Adept Lynx*. Der *Adept Lynx* ist ein *Autonomous Indoor Vehicle* – eine Untergruppe der *fahrerlosen Transportsysteme* (FTS). FTS können z. B. zum Warentransport innerhalb größerer Industrieanlagen eingesetzt werden. Im Gegensatz zu herkömmlichen FTS navigiert der *Adept Lynx* nicht spurgebunden, sondern autonom in einem zuvor eingelernten Gebiet. Zur Navigation und Kollisionsvermeidung besitzt er einen in Fahrtrichtung ausgerichteten Sicherheitslaser (SICK S300), der aus einem Schlitz in einer Höhe von 200 mm herauschaut (vgl. Abb. 1) [1]. Der Laser kann dank seines großen Öffnungswinkels von 270°, von denen allerdings nur 250° genutzt werden können, auch Hindernisse erkennen, die sich schräg hinter dem Roboter befinden.

Bevor der Roboter eingesetzt werden kann, muss der Arbeitsraum durch Abfahren mit Hilfe eines Joysticks kartiert werden, um die Orientierung zu gewährleisten. Der Arbeitsbereich (vgl. Abb. 6) kann durch eine Vielzahl verschiedener Modifikationen beeinflusst werden; zum Beispiel können Zielpunkte, verbotene Bereiche, bevorzugte Wege, Einbahnstraßen oder Bereiche mit Links-/Rechtsverkehr definiert werden. Nicht nur durch die zusätzlichen virtuellen Einflüsse ist die Lösung des Problems der Pfadplanung



Bild 1: Bild eines Adept Lynx

nicht trivial. Bei autonom navigierenden mobilen Robotern beeinflussen viele Parameter den zur Pfadplanung verwendeten Algorithmus. Sie sind stark abhängig von der Umgebung, in welcher der Roboter arbeitet, sodass für verschiedene Umgebungen unterschiedliche Parametersätze eine optimale Lösung liefern. Ein Pfad ist dabei ein zweidimensionaler Weg, welcher von der aktuellen Roboterposition zum angegebenen Zielpunkt führt. Der Roboter muss den Pfad zuerst planen und ihm dann bis zum Zielpunkt folgen.

Unser Ziel in diesem Paper ist es, einen geeigneten Algorithmus zu finden, der die Parameter der Pfadplanung und Pfadverfolgung optimiert. Da über die Struktur des Problems nicht viel bekannt ist, wenden wir uns der Gruppe der *Evolutionären Algorithmen* (EAs) zu. EAs haben den Vorteil, dass sie ohne Expertenwissen Probleme lösen können. Dabei generiert der EA durch sogenannte Variationsoperatoren Parametersätze, deren Güte durch einen Simulator berechnet werden kann. Der Simulator stellt damit eine *Black-Box* zur Berechnung von $x \mapsto f(x)$ dar. Die virtuelle Simulation hat den Vorteil, dass eine mögliche Kollision auf Grund eines schlechten Parametersatzes den Roboter nicht beschädigt. Die besten Parametersätze werden jeweils vom EA selektiert und in der nächsten Generation zur Erzeugung von zufällig modifizierten Varianten (sog. Nachkommen) verwendet.

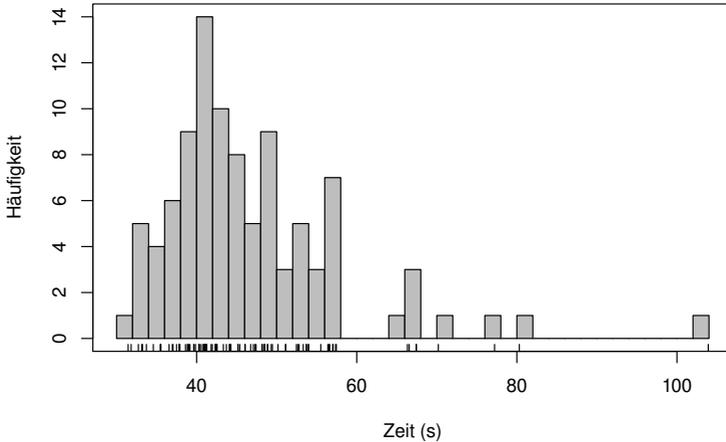


Bild 2: Auswertungsdauern von zufälligen Individuen auf einer Beispielkarte.

Da verschiedene, konfliktäre Zielfunktionen die Parameter bewerten, wird zur Optimierung ein *Evolutionärer Mehrkriterieller Optimier-Algorithmus* (EMOA) verwendet. Wir wählen den *S-Metrik-Selektion-EMOA* (SMS-EMOA) von Emmerich et al. [2]. Er hat gegenüber anderen EMOAs den Vorteil, auch bei mehr als zwei Zielfunktionen gute Lösungen zu finden [3]. Der erhöhte Rechenaufwand, den der SMS-EMOA im Vergleich zu anderen EMOAs zur Berechnung des Hypervolumenbeitrages [4] benötigt, kann vernachlässigt werden, da die Berechnung der Zielfunktion mittels Simulation deutlich zeitintensiver ist. Abbildung 2 zeigt die Dauer einiger Auswertungen für zufällig im Suchraum generierte Parametersätze. Die einzelnen Simulationszeiten zwischen 30 und 105 Sekunden sind abhängig vom Parametersatz und entstammen einem Parcours, welcher der Länge nach in etwa dem späteren Testparcours entspricht.

Grundlagen

Definition des Roboteroptimierungsproblems

Einer der wesentlichen Gesichtspunkte von mobilen Robotern ist die durchschnittliche Geschwindigkeit bzw. die Fahrzeit, denn ihr Haupteinsatzgebiet

ist der Warentransport. Je schneller Güter transportiert werden, z. B. vom Lager zur Maschine oder umgekehrt, desto höher ist der Durchsatz und desto mehr Fahrten können in einem bestimmten Zeitraum erfolgen. Für den Betrieb ergeben sich zwei Vorteile: Eine erhöhte Produktivität durch die gesteigerte Automatisierung und eine Kostenersparnis durch Senkung der Lohnkosten.

Darüber hinaus ist ein kollisionsfreier Transport unabdingbar. Kollisionen könnten z. B. mit Menschen entstehen, da der Roboter in deren Arbeitsraum eingesetzt werden kann. Um Verletzungen der Personen und Beschädigungen der vom Roboter transportierten Waren zu vermeiden, muss er Hindernisse so weiträumig umfahren, dass eine Kollision ausgeschlossen werden kann. Dazu darf eine gewisse Distanz zu Hindernissen nicht unterschritten werden.

Um die Güter besser zu schützen, soll der Roboter außerdem eine besonders sanfte Fahrweise haben, d. h. er darf nicht „stotternd“ fahren. Zum Beispiel beim Transport von Lithiumwafern in der Halbleiterindustrie können diese leicht beschädigt werden, wenn sie in ihren Boxen bei der Fahrt umher rutschen. Je weicher die Fahrweise ist, desto geringer ist die Gefahr, die transportierte Ware zu beschädigen. Außerdem beansprucht dies die Motoren der Antriebsräder nicht so stark, was sich wiederum positiv auf deren Lebensdauer auswirkt.

Die zu optimierenden Parameter bestehen im Wesentlichen aus der Genauigkeit der Pfadplanung, diversen Sicherheitsabständen des Roboters zu allen Seiten und Geschwindigkeits- und Beschleunigungswerten in verschiedenen Fahr Situationen. Insgesamt kommen so 16 Parameter zusammen, die als reellwertig behandelt werden können. Wie die Parameter das Verhalten genau steuern, wird von Menges [5] im Detail beschrieben.

Die drei genannten Kriterien können wie folgt numerisch erfasst werden:

- Die Fahrzeit ergibt sich aus der Differenz der Ankunfts- und Startzeit des Roboters: $t_{\text{Ziel}} - t_{\text{Start}}$. Sie bezieht sich jedoch nicht nur auf die reine Zeit in welcher der Roboter sich bewegt, sondern beinhaltet zudem die Zeit während der der Roboter noch auf der Stelle steht und den Pfad zum Ziel plant.
- Die Unruhe der Fahrweise kann ausgedrückt werden als die Summe der Geschwindigkeitsänderungen zu N diskreten Zeitpunkten. Dabei

kann zwischen translationalen und rotationalen Geschwindigkeitsänderungen unterschieden werden:

– Translational (Verschiebung): $\sum_{t=1}^{N-1} (v_{\text{trans},t} - v_{\text{trans},t+1})^2$

– Rotational (Drehung): $\sum_{t=1}^{N-1} (v_{\text{rot},t} - v_{\text{rot},t+1})^2$

Die Geschwindigkeiten sind hier jeweils mit v bezeichnet.

- Wenn X_t die Menge aller Distanzen zu Hindernissen zum Zeitpunkt t bezeichnet, so ergibt $\min\{\min(X_1), \min(X_2), \dots, \min(X_N)\}$ die kleinste Distanz zu irgendeinem Hindernis während der gesamten Fahrt.

Anstatt den minimalen Abstand zu Hindernissen als Zielfunktion zu verwenden, soll dieser als Restriktion gelten. Kein Hindernis darf näher als 50 mm zum Roboter sein. Sollte dies dennoch eintreten, wird der Parametersatz als unzulässig gekennzeichnet. Das hat zur Folge, dass die drei übrigen Zielfunktionen (Fahrzeit und die zwei Unruhewerte) auf einen vorab geschätzten Maximalwert gesetzt werden. Als eine Art Straffunktion wird zu diesem Maximalwert die Schwere der Verletzung der Grenze hinzu addiert. Damit wird der SMS-EMOA aus dem Bereich der unzulässigen Parameter heraus gedrängt, falls er diesen Bereich betritt. Die errechnete Distanz zum nächsten Hindernis kann dabei sogar negativ sein, was gleichbedeutend mit einer Kollision wäre, da das Hindernis in diesem Fall innerhalb des Roboters läge. Es ist außerdem möglich, dass die Lösung aus Parametern besteht, die es dem Roboter nicht ermöglichen sich in der Umgebung zu bewegen. In diesem Fall werden die Werte der Zielfunktionen auf die Summe des maximalen Straffunktionswertes und des Maximalwertes gesetzt.

Der asynchrone SMS-EMOA

Unsere Anwendung lässt zur Problemlösung eine parallele Auswertung der Individuen zu. Eine (synchronisierte) Parallelisierung der Auswertung durch Anhebung der Anzahl von Nachkommen auf $\lambda > 1$ würde jedoch (bei exakter Berechnung) zu einem exponentiell erhöhten Aufwand bei der Bestimmung des Hypervolumenbeitrages im SMS-EMOA führen. Außerdem resultieren große Differenzen in der Evaluationszeit einzelner Individuen

in Wartezeit auf die zeitintensivste Auswertung, da erst nach der Bewertung aller Nachkommen neue Individuen generiert werden können. Diese Leerlaufzeiten verschwenden Ressourcen [6]. Abbildung 3 (links) zeigt ein beispielhaftes Verhältnis zwischen Arbeits- und Leerlaufzeit. Die Zeit, in der ein Prozess einen Funktionswert berechnet, ist durch eine horizontale schwarze Linie dargestellt; gestrichelte horizontale Linien sind Leerlaufzeiten der Prozesse. Dieses Beispiel zeigt die Auswertung von vier Generationen eines parallelisierten $(\mu + 6)$ -EA. Die sechs Nachkommen werden von den Prozessen parallel ausgewertet. Die durchgezogenen vertikalen Linien kennzeichnen die Wechsel von einer Generation zur nächsten. An dieser Stelle müssen die einzelnen Auswertungsprozesse miteinander synchronisiert werden; dort entsteht der Zeitverlust. Die Abbildung veranschaulicht, dass 29 der insgesamt 84 Zeitschritte (6 Prozesse zu je 14 Zeitschritten) ungenutzt bleiben. Die Prozesse verbringen damit knapp 35% ihrer Zeit im Leerlauf.

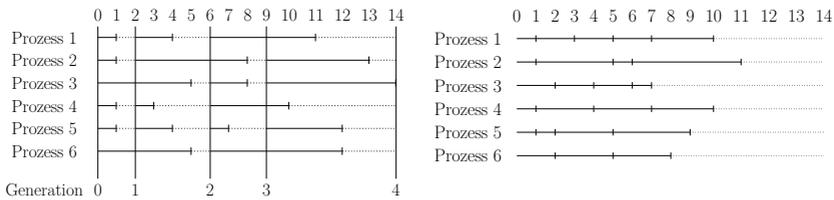


Bild 3: Beispielhafter Vergleich von synchroner (links) und asynchroner Parallelisierung (rechts). Im asynchronen Fall können unter Verzicht auf das Generationenkonzept drei Zeitschritte eingespart werden.

Wir implementieren den SMS-EMOA stattdessen als asynchronen Algorithmus, damit diese Leerlaufzeit minimiert wird. Die Idee ist, dass mehrere Prozesse eine gemeinsame Population verbessern. Das Selektionsschema des asynchronen SMS-EMOA bleibt ein $(\mu + 1)$ Schema. Sobald ein Prozess die Auswertung abgeschlossen hat, bindet er das bewertete Individuum direkt in die aktuelle Population ein bzw. verwirft es, falls das Individuum keine Verbesserung bringt. Sollte das Abbruchkriterium noch nicht erfüllt sein, wird anschließend das nächste Individuum aus der aktualisierten Population generiert und wiederum bewertet. Jeder einzelne Prozess führt somit alle Aufgaben eines SMS-EMOA aus (vgl. Algorithmus 2) – nicht nur die Evaluation wie im parallelen Fall.

Algorithmus 1.: Initialisierung des asynchronen SMS-EMOA

- 1: $P \leftarrow \text{allocate}(\mu)$ ▷ Initialisiere Platzhalter für μ Individuen
 - 2: $tp \leftarrow \text{initThreadPool}(\tau)$ ▷ Initialisiere Threadpool mit τ Threads
 - 3: $t \leftarrow 0$ ▷ Generations-Zähler
 - 4: $\text{mutex} \leftarrow \text{initMutex}()$ ▷ Initialisiere Mutex zur Synchronisation
 - 5: $\text{start}(tp)$ ▷ Starte asynchrone Ausführung des SMS-EMOA
-

Algorithmus 2.: Hauptteil des asynchronen SMS-EMOA

- 1: $P_0 \leftarrow \text{init}(\text{mutex})$ ▷ Beteiligung an Erstellung der initialen Population
 - 2: **while** Abbruchkriterium nicht erfüllt **do**
 - 3: $\mathbf{x} \leftarrow \text{createOffspring}(P_t, \text{mutex})$ ▷ Erzeuge 1 Nachkommen
 - 4: $\text{evaluate}(\mathbf{x})$ ▷ Berechne Zielfunktionswerte des Nachkommen
 - 5: $\text{enter}(\text{mutex})$ ▷ Verriegle Population für andere Prozesse
 - 6: $Q_t \leftarrow P_t \cup \{\mathbf{x}\}$
 - 7: $\{F_1, \dots, F_w\} \leftarrow \text{fastNondominatedSort}(Q_t)$ ▷ Sortiere in w Fronten
 - 8: $\mathbf{r} \leftarrow \text{createReferencePoint}(F_w)$ ▷ Berechne Referenzpunkt zur Front F_w
 - 9: $\mathbf{x}^* \leftarrow \text{argmin}_{\mathbf{x} \in F_w}(\Delta_s(\mathbf{x}, F_w, \mathbf{r}))$ ▷ Bestimmte \mathbf{x}^* mit kleinstem Beitrag
 - 10: $P_{t+1} \leftarrow Q_t \setminus \{\mathbf{x}^*\}$ ▷ Entferne schlechtestes Element
 - 11: $t \leftarrow t + 1$
 - 12: $\text{leave}(\text{mutex})$ ▷ Gebe Population für andere Prozesse frei
-

Algorithmus 1 beschreibt die globale Initialisierung des asynchronen SMS-EMOA. Er legt die geteilte Population mit Speicherplatz für μ Individuen, die einzelnen Prozesse und ein Mutex zu deren Synchronisation an. Außerdem initialisiert er den Generationszähler und startet die asynchrone Ausführung. Jeder gestartete Prozess führt den Code aus Algorithmus 2 aus. Die „init“ genannte Methode in Zeile 1 erzeugt dabei solange zufällig gleichverteilte Individuen, wertet sie aus und fügt sie der Population hinzu, wie die Populationsgröße μ noch nicht erreicht ist. Die Population könnte auch vor dem Start der einzelnen Prozesse mit μ Individuen sequentiell initialisiert werden, dabei käme allerdings der Vorteil der asynchronen Evaluation bei der initialen Population nicht zum Tragen.

Die Prozesse greifen auch bei der Auswahl der Elter-Individuen für die Erzeugung von Nachkommen lesend auf die Population zu. Deshalb muss die Methode `createOffspring` dafür sorgen, dass die Population währenddessen nicht verändert wird (vgl. Algorithmus 3).

Algorithmus 3.: createOffspring(population, mutex)

```
1: enter(mutex)           ▷ Verriegle Population für andere Prozesse
2: parents ← select(population)   ▷ Selektiert die Elter-Individuen
3: leave(mutex)          ▷ Gebe Population für andere Prozesse frei
4: child ← crossover(parents)     ▷ Rekombination
5: child ← mutate(child)         ▷ Mutation
6: return child
```

Experimente

Vergleich mit sequentieller Variante auf Testproblemen

Um sicherzugehen, dass unser neuer Algorithmus von der Approximationsgüte der Pareto-Front her nicht zu stark abfällt, vergleichen wir ihn zunächst mit dem herkömmlichen SMS-EMOA ohne Parallelisierung.

Vorüberlegungen Es ist zu erwarten, dass die Güte der Lösungen des asynchronen SMS-EMOA von den Lösungen des unveränderten SMS-EMOA übertroffen wird. Durch die simultane Auswertung von k Individuen bei der asynchronen Variante ist allerdings damit zu rechnen, dass die Laufzeit der Standardvariante um das k -fache länger ist. Daher führen wir den Vergleich auf künstlichen Testproblemen durch.

Aufgabenstellung Die Experimente sollen einen möglichen Leistungsverlust für die asynchrone Variante in Bezug auf die Güte der gefundenen Lösungen untersuchen. Die Ergebnisse werden anhand des Hypervolumen-Indikators, des additiven Epsilon-Indikators [7], der *generational distance* (GD) und der *inverted GD* (IGD) [8] bewertet. Zusätzlich werden die Laufzeiten der Algorithmen gemessen.

Setup Beide Algorithmen verwenden *Simulated Binary Crossover* und polynomielle Mutation mit Distributionsindizes von 20 zur Variation. Die Anwendungswahrscheinlichkeiten für Rekombination und Mutation pro Dimension sind 0,9 respektive 1. Der asynchrone SMS-EMOA arbeitet mit fünf Prozessen/Threads gleichzeitig auf der gemeinsamen Population. Der Offset zur Konstruktion des Referenzpunktes für die Selektion beträgt

immer 100. Beide Varianten bekommen das gleiche Budget von 10^4 Auswertungen und eine Populationsgröße von 40. Dominierte Lösungen werden in der Auswertung grundsätzlich ignoriert.

Als Testprobleme kommen ZDT1, ZDT3, DTLZ1 und DTLZ7 zum Einsatz [9, 10]. Die für das Hypervolumen benötigten Referenzpunkte sind $(2, 2)^\top$ für die ZDT-Probleme und $(8, 16, 3)^\top$ und $(2, 2, 8)^\top$ für DTLZ1 respektive DTLZ7. Des Weiteren haben die ZDT-Probleme 30 Entscheidungsvariablen, die DTLZ-Probleme jedoch nur sieben, um dem höheren Schwierigkeitsgrad durch die drei Zielfunktionen Rechnung zu tragen.

Die Funktionsauswertung dieser Probleme benötigt kaum Rechenzeit. Um zeitintensive Auswertungen abzubilden, werden die Testfunktionen um eine Wartezeit erweitert. Diese beträgt zwischen zwei und vier Sekunden, um auch zeitliche Unterschiede zwischen den Auswertungen zu simulieren. Die zufällige Generierung der Wartezeit geschieht unabhängig von den Entscheidungsvariablen des Individuums. Das entspricht im Vergleich zur Optimierung der Pfadplanung nicht der Realität, da die benötigte Simulationszeit maßgeblich von den Entscheidungsvariablen abhängt und dazu eine der Zielfunktionen darstellt. Dennoch sollte die zufällige Verlängerung der Auswertung eine Tendenz im Bezug auf die Zeitersparnis erkennen lassen. Algorithmen und Testprobleme sind in Java mit Hilfe der jMetal-Bibliothek implementiert [11]. Wegen der Zufallseinflüsse wird jeder Lauf zehnmal wiederholt.

Ergebnisse Abbildungen 4 und 5 zeigen repräsentative Approximationsmengen der beiden Algorithmen im Zielfunktionsraum. Tabelle 1 enthält gemittelte Indikatorwerte über die zehn Läufe plus/minus des Standardfehlers. In [5, S. 57–64] werden die Ergebnisse in detaillierterer Form mit weiteren Indikatoren dargestellt.

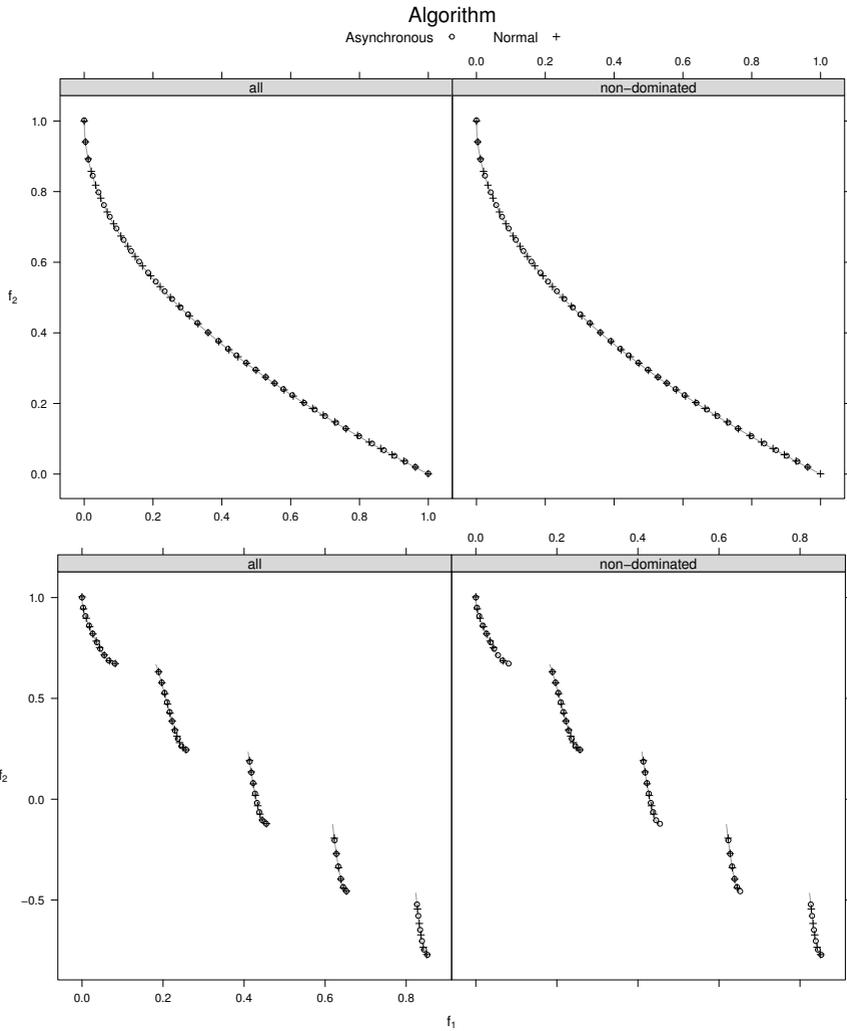


Bild 4: Vergleich der nicht-dominierten Front des asynchronen SMS-EMOA und der Standardvariante auf ZDT1 (oben) und ZDT3 (unten). Die grauen Linien zeigen die wahre Pareto-Front.

Algorithm

Asynchronous ◦ Normal +

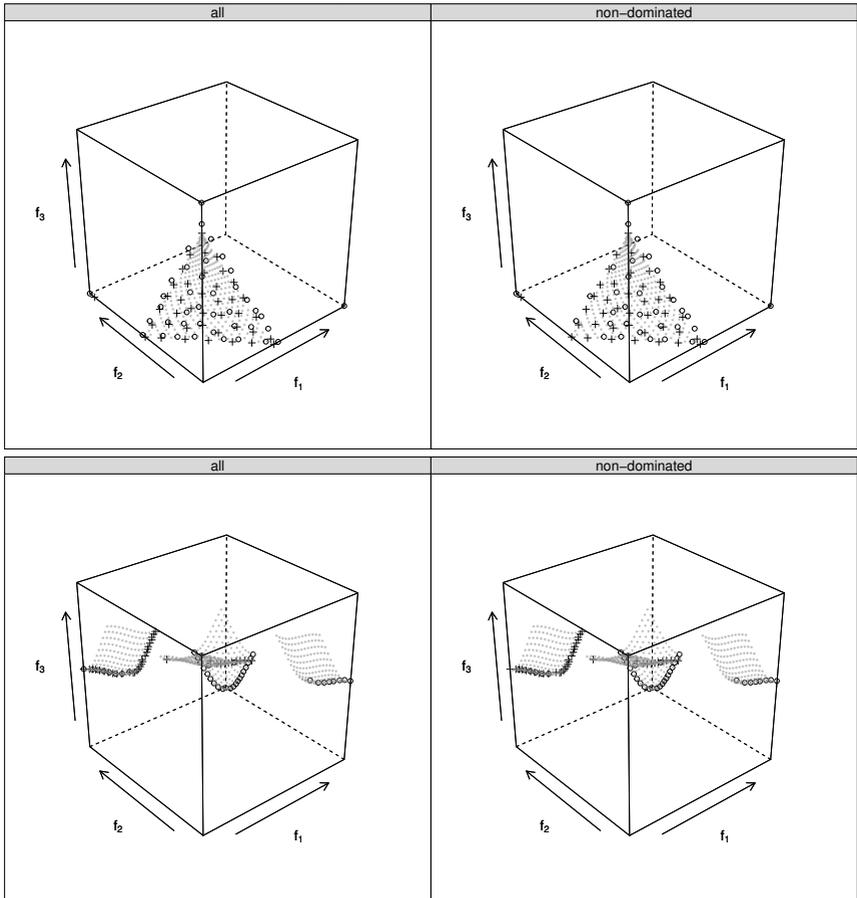


Bild 5: Vergleich der nicht-dominierten Front des asynchronen SMS-EMOA und der Standardvariante auf DTLZ1 (oben) und DTLZ7 (unten). Die grauen Punkte stellen eine Referenzmenge von pareto-optimalen Lösungen dar.

Problem	Indikator	Algorithmus	
		Asynchron	Original
ZDT1	Hypervolumen	$3,6516 \pm 0,0003$	$3,6495 \pm 0,0028$
	Epsilon	$0,0155 \pm 0,0045$	$0,0170 \pm 0,0014$
	GD	$0,000\,201 \pm 1,5 \cdot 10^{-5}$	$0,000\,172 \pm 8,3 \cdot 10^{-6}$
	IGD	$0,000\,347 \pm 1,3 \cdot 10^{-6}$	$0,000\,355 \pm 1,0 \cdot 10^{-5}$
	Dauer (s)	6009 ± 3	29994 ± 12
ZDT3	Hypervolumen	$4,7344 \pm 0,0461$	$4,7644 \pm 0,0345$
	Epsilon	$0,0716 \pm 0,0388$	$0,0583 \pm 0,0317$
	GD	$0,000\,183 \pm 1,6 \cdot 10^{-5}$	$0,000\,654 \pm 4,5 \cdot 10^{-4}$
	IGD	$0,000\,731 \pm 2,9 \cdot 10^{-4}$	$0,000\,703 \pm 2,7 \cdot 10^{-4}$
	Dauer (s)	5999 ± 3	30042 ± 24
DTLZ1	Hypervolumen	$383,823 \pm 0,075$	$383,798 \pm 0,081$
	Epsilon	$0,161 \pm 0,043$	$0,204 \pm 0,060$
	GD	$0,0854 \pm 0,0242$	$0,1435 \pm 0,0362$
	IGD	$0,0036 \pm 0,0012$	$0,0037 \pm 0,0012$
	Dauer (s)	6009 ± 4	30075 ± 16
DTLZ7	Hypervolumen	$16,26 \pm 0,60$	$15,44 \pm 0,44$
	Epsilon	$0,54 \pm 0,23$	$0,84 \pm 0,16$
	GD	$0,0018 \pm 5,5 \cdot 10^{-5}$	$0,0019 \pm 5,4 \cdot 10^{-5}$
	IGD	$0,0101 \pm 0,0017$	$0,0130 \pm 0,0014$
	Dauer (s)	6007 ± 2	30040 ± 14

Tabelle 1: Ergebnisse auf den Testproblemen.

Beobachtungen Grundsätzlich sind die Approximationsmengen der beiden Algorithmen optisch kaum voneinander zu unterscheiden. Bei DTLZ7 fällt auf, dass beide Algorithmen es nicht schaffen, die ganze Front abzudecken, sondern nur Punkte an bestimmten Rändern anbieten. Auch die Unterschiede in Tab. 1 sind in der Regel nicht signifikant – natürlich mit Ausnahme der Laufzeiten. Diese unterscheiden sich wie erwartet ziemlich genau um den Faktor fünf.

Diskussion Die Experimente zeigen, dass die asynchrone Variante in der gewählten Konfiguration von der Leistung her nicht unterscheidbar ist vom originalen, sequentiellen Algorithmus. Da die Auswertungsdauer der Zielfunktionen künstlich zum Flaschenhals der Berechnungen gemacht wurde, so wie dies auch in der realen Welt zu erwarten ist, wirkt sich hingegen die asynchrone Parallelisierung nahezu perfekt auf die Laufzeit aus.

Parameteroptimierung eines mobilen Roboters

In diesem Abschnitt vergewissern wir uns der Leistungsfähigkeit des asynchronen SMS-EMOA, indem wir ihn für die Parameteroptimierung eines mobilen Roboters einsetzen.

Vorüberlegungen Als Vorbereitung zur Optimierung wurden die Parameter gleichverteilt über ihre zulässigen Wertebereiche verändert. Dabei kam heraus, dass besonders die Zielfunktion der Glätte translational über einen großen Wertebereich verfügt. Dadurch können Verbesserungen dieser Zielfunktion einen großen Anteil am dominierten Hypervolumen einnehmen. Um die Randpunkte bei der Selektion der nächsten Generation beizubehalten, wird für die Referenzpunktberechnung des SMS-EMOA hier ein Offset von 10^5 gewählt.

Aufgabenstellung Wir erwarten, dass der asynchrone SMS-EMOA eine Verbesserung hinsichtlich der Qualität darstellt, im Vergleich mit einem Ansatz, der das Budget gleichmäßig auf eine Anzahl von unabhängigen SMS-EMOA-Instanzen verteilt. Die Qualität wird mittels des Indikators Hypervolumen gemessen.

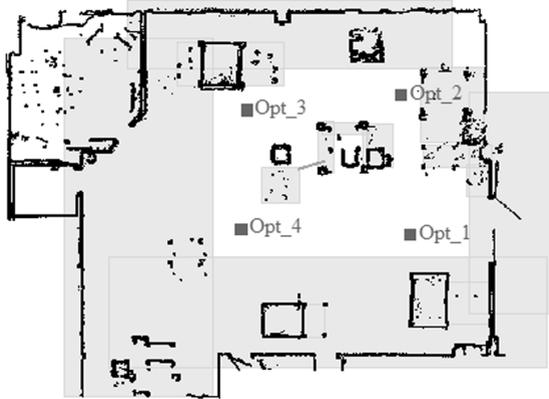


Bild 6: Karte auf der die Pfadplanungsparameter optimiert werden.

Setup Der Grad der Parallelisierung ergibt sich aus der zur Verfügung stehenden Rechenleistung für die Simulation und beträgt in diesem Experiment 16. Trotz der zeitintensiven Zielfunktionsauswertungen bekommen die Algorithmen ein hohes Budget von 160000 Auswertungen. Im Falle der parallel ausgeführten einzelnen SMS-EMOAs wird das Budget gleichmäßig verteilt. Die Population der asynchronen Variante besteht aus 320 Individuen, woraus sich eine Populationsgröße von 20 Individuen bei den parallel ausgeführten SMS-EMOAs ergibt.

Die Optimierung findet auf einer Karte vom Applikationslabor Adept Dortmund statt, welche in Abbildung 6 zu sehen ist. Dadurch, dass die Parameter auf einer realen Karte optimiert werden, können die Lösungen anschließend mit dem echten Roboter auf ihr verifiziert werden. Die Karte besteht aus den vier Punkten Opt_1 bis Opt_4 , die nacheinander in einer Route abgefahren werden. Der Roboter startet auf einer definierten Position rechts oberhalb von Opt_1 . An diesem Punkt wird der neue Parametersatz für das nächste Experiment eingestellt und die Route gestartet.

Der Bereich in dem der *Adept Lynx* fährt, ist durch verbotene Zonen eingeschränkt. Die kürzeste und schnellste Route ist direkt von Opt_1 nach oben zu Opt_2 zu fahren. Diese Strecke verfügt über eine enge Passage zwischen zwei verbotenen Zonen hindurch. Je nach Parametersatz kann es

Algorithmus	Hypervolumen	Lösungen	I_{NDC}	Laufzeit
Asynchron	$17,872 \cdot 10^{12}$	318	317	168,62 h
Parallel	$17,261 \cdot 10^{12}$	88	9	267,80 h

Tabelle 2: Vergleich von Hypervolumen, Anzahl nicht-dominiertes Lösungen und Laufzeit der Lösungen der Pfadplanung.

dazu kommen, dass der *Adept Lynx* einen Umweg in Kauf nehmen muss, weil auf Grund der angenommenen Roboterbreite eine Pfadplanung durch die Passage fehlschlägt. In diesem Fall fährt er nicht gerade nach oben, sondern erreicht den zweiten Wegpunkt *Opt_2* nur, indem er außen herum *Opt_4* und *Opt_3* passiert. Diese Wegpunkte gelten selbstverständlich nicht als erreicht, sondern der *Adept Lynx* muss nach Erreichen von *Opt_2* die restlichen Punkte anfahren. Falls ein Fahrzeit-optimierter Parametersatz es dem Roboter nicht ermöglichen sollte, den direkten Weg zu nehmen, wäre die Optimierung gescheitert.

Ergebnisse Die Tabelle 2 zeigt einige Indikatoren zur Bewertung der Lösungen. Die Abbildung 7 zeigt die Ergebnisse der physikalischen Experimente im Vergleich zu den Lösungen der Simulation.

Beobachtungen I_{NDC} in Tabelle 2 gibt an, wie hoch der Beitrag zu einer gemeinsamen Menge nicht-dominiertes Lösungen ist. Die Werte zeigen, dass eine der 318 nicht-dominierten Lösungen der asynchronen Variante von Lösungen der parallelen SMS-EMOAs dominiert wird. Es gibt hingegen nur neun Lösungen der parallelen SMS-EMOAs, die nicht von den Asynchronen dominiert werden. Überhaupt ergibt die Vereinigung der Populationen der parallelen Varianten nur 88 nicht-dominierte Individuen. Die Laufzeit der asynchronen Variante liegt zudem deutlich unter der Laufzeit der parallel ausgeführten Algorithmen. Sie benötigt zur Optimierung der Parameter 168,62 Stunden und damit ca. 100 Stunden weniger als die parallelen SMS-EMOAs.

Abbildung 7 zeigt, dass die Lösungen beim Test auf dem realen Roboter ähnlich verteilt sind wie die Lösungen der Simulation. Die tatsächliche Fahrzeit liegt leicht über der Fahrzeit der Simulation und kann recht zuverlässig optimiert werden. Bei den Zielfunktionen zur Optimierung der Laufruhe

Results on real Robot

Source

Default ○ Optimized +

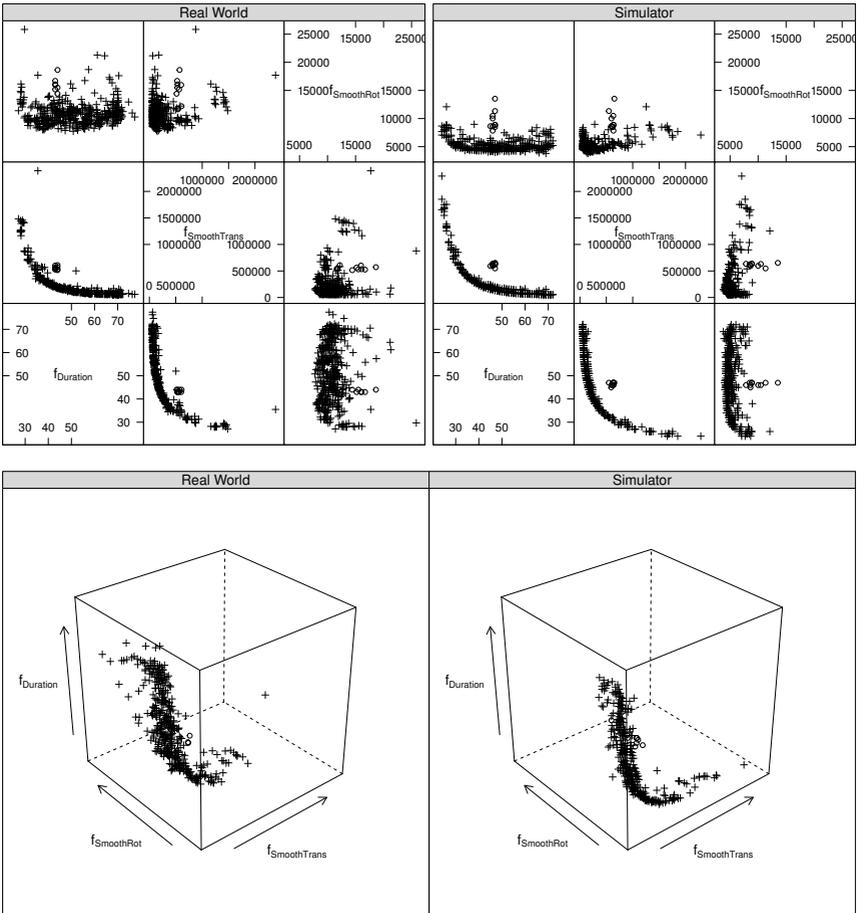


Bild 7: Vergleich der optimierten Lösungen mit der Standardkonfiguration auf dem realen Roboter und der Simulation als Streudiagrammmatrix und als 3D-Punktewolke.

weichen die Werte stärker von den simulierten Ergebnissen ab. Ganz besonders stark tritt dies bei der rotatorischen Laufruhe auf. Dort verdoppeln sich die Werte. Die gegensätzliche Wechselbeziehung zwischen f_{Duration} und $f_{\text{SmoothTrans}}$ ist auch beim realen Roboter gut zu erkennen. Außerdem ist anhand der Skalierung der Grafiken zu erkennen, dass der echte *Adept Lynx* deutlich stärker oder häufiger rotiert als der simulierte Roboter. Ferner ist in der Abbildung zu erkennen, dass es optimierte Lösungen gibt, die in allen Zielfunktionen echt besser sind als die Standardkonfiguration. Nach Ausfiltern von dominierten Lösungen bleiben 150 Lösungen übrig, die echte Verbesserungen gegenüber der Standardkonfiguration sind.

Diskussion Die Differenz in der Laufzeit in Tabelle 2 ist überraschend und entsteht vermutlich dadurch, dass die Auswertungsdauer gleichzeitig auch Zielfunktion ist. Somit haben Fokussierungen auf unterschiedliche Teile des Suchraums durch die großen Unterschiede in der Auswertungsdauer einzelner Parametersätze auch großen Einfluss auf die Gesamtlaufzeit. Normalerweise wäre die Ausführung unabhängiger Algorithmenläufe hinsichtlich der Parallelisierung eigentlich optimal, da überhaupt kein Kommunikationsaufwand entsteht.

Die Zielfunktion zur rotatorischen Laufruhe scheint sich nicht sonderlich zuverlässig per Simulation ermitteln zu lassen. Anscheinend weicht der Roboter Hindernissen in der realen Welt anders aus als im Simulator. Das kann daran liegen, dass die Unsicherheit in der realen Welt deutlich höher ist und der Fahrweg häufiger korrigiert werden muss. Dennoch kann der Simulator zur Parameteroptimierung verwendet werden. Die Lösungen müssen allerdings nach der Optimierung auf dem *Adept Lynx* getestet werden.

Fazit

Diese Arbeit demonstrierte die mehrkriterielle evolutionäre Parameteroptimierung eines mobilen Roboters mittels des SMS-EMOA. Die Ergebnisse zeigen, dass der entwickelte asynchrone Algorithmus die Leistung unabhängig voneinander parallel ausgeführter SMS-EMOAs übertrifft und auf den verwendeten Testproblemen keine großen Qualitätsverluste gegenüber dem unveränderten Algorithmus hinnehmen muss. Er reduziert damit die

Laufzeit bei Problemen, für die eine parallele Auswertung möglich ist, ohne signifikanten Qualitätsverlust. Dank eines vorhandenen Simulators ist die Parameteroptimierung des mobilen Roboters *Adept Lynx* ein solches Problem. Die Parameter zur Pfadplanung wurden anhand von drei konfliktären Kriterien optimiert. Auch die Ergebnisse zur Pfadplanung des *Adept Lynx* zeigen eine deutliche Verbesserung gegenüber der Standardkonfiguration. Für einen Produktiveinsatz der Parameteroptimierung müsste allerdings eine größere Anzahl von Routen pro Karte abgefahren werden, um eine Überanpassung an einzelne Strecken zu verhindern.

Da eine synchron parallelisierte Variante des SMS-EMOA konzeptuelle Nachteile hat, wurde sie hier nicht untersucht. In Zukunft sollten die Unterschiede des asynchronen SMS-EMOA zur synchron parallelisierten Variante jedoch auch experimentell quantifiziert werden.

Abschließend ist zu bemerken, dass das hier vorgestellte Konzept der asynchronen Parallelisierung unabhängig von evolutionären Algorithmen und mehrkriterieller Optimierung ist und daher auf viele andere Optimierungsalgorithmen übertragen werden kann.

Literatur

- [1] Banwell, P.: *Adept Lynx Platform User's Guide*. Adept Technology, Inc., 5960 Inglewood Drive - Pleasanton, CA 94588 - USA, 11970-000 rev d Aufl. URL http://www1.adept.com/main/KE/DATA/Mobile/Lynx_UG.pdf. 2015.
- [2] Emmerich, M.; Beume, N.; Naujoks, B.: An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In: *Evolutionary Multi-Criterion Optimization* (Coello Coello, C.; Hernández Aguirre, A.; Zitzler, E., Hg.), Bd. 3410 von *Lecture Notes in Computer Science*, S. 62–76. Springer. 2005.
- [3] Beume, N.; Naujoks, B.; Rudolph, G.: SMS-EMOA - Effektive evolutionäre Mehrzieloptimierung (SMS-EMOA - Effective Evolutionary Multiobjective Optimization). *Automatisierungstechnik* 56 (2008) 7, S. 357–364. URL <http://dx.doi.org/10.1524/auto.2008.0715>.

- [4] Zitzler, E.; Thiele, L.: Multiobjective optimization using evolutionary algorithms — A comparative case study. In: *Parallel Problem Solving from Nature — PPSN V* (Eiben, A. E.; Bäck, T.; Schoenauer, M.; Schwefel, H.-P., Hg.), Bd. 1498 von *Lecture Notes in Computer Science*, S. 292–301. Springer. 1998.
- [5] Menges, D. A.: *Parameteroptimierung des mobilen Roboters Adept Lynx*. Diplomarbeit, Fakultät für Informatik, Technische Universität Dortmund. 2015.
- [6] Kolda, T. G.; Torczon, V. J.: Understanding Asynchronous Parallel Pattern Search. In: *High Performance Algorithms and Software for Nonlinear Optimization* (Di Pillo, G.; Murli, A., Hg.), Bd. 82 von *Applied Optimization*, S. 323–342. Springer. 2003.
- [7] Zitzler, E.; Knowles, J.; Thiele, L.: Quality Assessment of Pareto Set Approximations. In: *Multiobjective Optimization* (Branke, J.; Deb, K.; Miettinen, K.; Słowiński, R., Hg.), Bd. 5252 von *Lecture Notes in Computer Science*, S. 373–404. Springer. 2008.
- [8] Coello Coello, C. A.; Cruz Cortés, N.: Solving Multiobjective Optimization Problems Using an Artificial Immune System. *Genetic Programming and Evolvable Machines* 6 (2005) 2, S. 163–190.
- [9] Zitzler, E.; Deb, K.; Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation* 8 (2000) 2, S. 173–195.
- [10] Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E.: Scalable Test Problems for Evolutionary Multi-Objective Optimization. Techn. Ber. 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH, Gloriastrasse 35., ETH-Zentrum, CH-8092, Zürich, Switzerland. 2001.
- [11] Durillo, J. J.; Nebro, A. J.: jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software* 42 (2011), S. 760–771.

Evolution of Controllers for Robot-Plant Bio-Hybrids: A Simple Case Study Using a Model of Plant Growth and Motion

Mostafa Wahby¹, Mohammad Divband Soorati¹,
Sebastian von Mammen², Heiko Hamann¹

¹Heinz Nixdorf Institute, Department of Computer Science,
University of Paderborn, Paderborn, Germany
E-Mail: mostafa.wahby@uni-paderborn.de

²Organic Computing, University of Augsburg, Augsburg, Germany
E-Mail: sebastian.von.mammen@informatik.uni-augsburg.de

Introduction

In evolutionary robotics methods of evolutionary computation are applied to evolve robot controllers [1]. Evolutionary robotics is also our method of choice in the EU-funded project *flora robotica* [6, 8], which pushes research towards the evolution of a broad variety of artifacts and contraptions [4]. We are investigating how a distributed robot system and a group of biological plants can be tightly coupled to generate synergies and finally result in a bio-hybrid system. We want to create a co-dependent and self-organized system with closely linked symbiotic relationships where plants support robots, for example, by providing scaffolding and robots that direct plant growth towards desired areas. In addition to plant growth there is also plant motion which is often ignored, probably due to the fact that the motion of plants is slow compared to that of animals. It can be difficult sometimes to distinguish between a plant's motion and growth as both happen concurrently and usually in similar ways. They can differ, however, fundamentally in terms of time scales. Another important difference is the fact that motion can be reverted while growth is mostly permanent. Besides plant growth we also want to harness plant motion in our robot-plant systems.

In evolutionary robotics, the evaluation of an individual's fitness is a critical and often costly task (time, wear of robots, etc.). This is true for both options, evaluating robot controllers in simulation or via so-called embodied evolution with actual robots in hardware [17]. The potential speed-up of using simulations is diminished by the so-called reality gap problem [9]. Robot controllers evolved in simulation might perform poorly in reality due to imperfectly simulated features of reality. Still, the evaluation of a candidate controller might be performed within a matter of seconds. For example, in embodied evolution of an object avoidance controller the feedback of hitting an object can be instantly obtained from the robot's sensors and would immediately indicate a failure.

In this work, we apply artificial evolution to robot-plant bio-hybrid systems. An intuitive and simple task in this context is controlling plant growth and motion towards light (phototropism). Control here means to direct the plant tip to a certain position by switching lights on and off. It is obvious that in the case of the embodied evolution approach the feedback from the plants during an evaluation would be very slow. For example, the common bean plant (*Phaseolus vulgaris*), which is actually considered a fast grower, grows an average of 2cm to 3cm per day during an early growth stage. Hence, an evaluation period of at least three days would be required to grow the plant tip to reach a target initially 9cm away. Instead, we consider the option to priorly control plant motion. According to our experiments, bean plants bend towards a light source with a velocity of up to 4.4mm/min. This is considered relatively quick feedback and sufficient for evolving controllers in an embodied approach. Accordingly, the objective of our experiments is to maximize the plant motion within a given period of time by switching between different light settings. The resultant plant movement behavior could be useful, for instance, as a signal to draw the attention of passers-by in public spaces or to implement facades that dynamically provide visual protection. In this paper, we present our approach to create a simple growth-model of the common bean based on empirically obtained data. We then use the model as a simulation to evolve closed-loop controllers that maximize the plant's motion and grow the plant's tip to three different targets. For the future, we plan to investigate how well these evolved controllers work on real plants (reality gap), means of embodied evolution for plant motion control, and augmented user interfaces to effectively design and utilize bio-hybrid systems.

Related work and context of this research

Besides the above mentioned relation to evolutionary robotics [1] and embodied evolution of things [17, 4], our work is also related to efforts in plant modeling. In this paper we present an early preliminary plant modeling approach focused on the plant tip only. There is, however, a straightforward, simple incremental approach to extend the model by separating the plant logically into several segments. That is the idea of organizing the individual segments of a virtual plant stem in transformation hierarchies. This approach is similar to preceding works, for instance, for retracing motion dynamics of trees due to wind gusts [19]. As a result of the hierarchical organization, changes introduced to one segment of the plant affect all those segments lower in the hierarchy, or closer to the tip of the plant. In L-systems [10], which represent a wide-spread approach to model plant growth, a set of production rules is iteratively applied (in parallel) to subsume symbols of a character string which, in turn, is geometrically interpreted at each step of the simulation. The geometry of a tree and its variation have recently been inversely computed from polygon meshes of actual trees and derived abstract branching trees [19, 13]. Allowing the individual nodes in such an abstract tree to act as autonomous, reactive agents [18]. These agents, in turn, might be attracted to light sources, which provides a simple approach to model dynamic plants that react to environmental stimuli. To the knowledge of the authors, swarm grammars represent the first such implementation that also integrated the branching behavior of L-systems [15]. In fact, phototropism and reactions with the environment were among the first experiments conducted with this agent-based L-system extension. Considering the stiffness of each of the connected stem elements of the virtual plant in combination with the urge towards a light source, we can efficiently approximate the bending motion of the plant [5].

The plant growth–motion model and the evolution of controllers presented in this paper are an effort within the interdisciplinary project *flora robotica*. The project creates and investigates mixed societies of robots and natural plants and brings together scientists from plant science, architecture, zoology, robotics, and computer science. The general idea to automate gardening tasks has been addressed before [3]. Also the automation of agriculture has been studied [11]. However, our objectives go beyond the

idea of mere automation. Our motivation in *flora robotica* is to develop a bio-hybrid system that assigns equal roles to plants and robots and creates synergies between them.

A natural plant can grow structures and sense environmental features, while robots can impose artificial stimuli on the plant and add to the plant's sensing and decision-making capabilities. That way we use robots to trigger artificial growth. The idea is to leverage natural adaptive behavior in plants [7] to extend the capabilities of robots and to leverage the free programmability of robots to create artificial growth processes. Our main idea is to extend the rich variety of natural growth processes [14] with artificial growth processes. There are many potential applications but we focus on the artificial growth of potentially dynamic architectural artifacts.

The research presented in this paper is a machine learning approach to create robot controllers that influence the growth of plants. It is an initial step towards the creation of a complex bio-hybrid system, i.e. a decentralized, self-organizing multi-robot system tightly coupled with plants.

Bean growth and motion experiments

We have created a setup that allows us to conduct simple experiments with plants in order to study and explore the possibilities of controlling their growth and motion. The choice for a specific plant species was driven by the facts that (a) the speed of growth and motion is of great concern, and (b) the plant needed to grow in standard robot lab conditions and office areas. A good compromise to cope with these constraints is the common bean plant (*Phaseolus vulgaris*) in its early growth stage. In our setup, as soon as the bean plant starts sprouting, we place it inside a box of 2m height and 1,20m in width and breath, clad in black cloth from the inside to reduce light reflections and to allow for taking high contrast photos. Next, we impose a light stimulus on the plant using two light sources located 30cm above the plant, and 30cm to the left and right, respectively (see Fig. 1(a)). They are turned on alternately every six hours for a total period of a 72 hours.

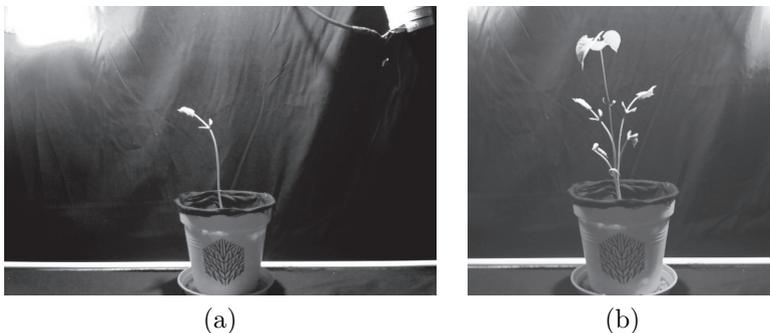


Figure 1: Simple growth and motion experiment: (a) Light from the left and right is alternatingly shone on a common bean plant. (b) Superposition of states indicating the amplitude of the plant's growth and left-right motion.

The light sources are two Adafruit NeoPixel RGB LED strips with 144 LEDs each. In our setting, each strip emits white light at full brightness which requires a current of up to 8.64A. Each LED has a power consumption of 0.24W and emits 18 lumen. A Raspberry Pi is used to operate the LED strips and a camera module¹ that takes a picture every five minutes, resulting in 864 images for each experiment, and 5184 images in total.

In Fig. 2 we show a sequence of photos from one of six experiments. The 16 photos depict growth and movement throughout an initial time period of 48 hours, with three hours between the shots². After 48 hours the plant has grown to about 20cm. During the six-hour activations of each light source, the plant bends towards it while maintaining a counterclockwise turning behavior (standard climbing behavior of bean plants, not seen in the photos). In Fig. 1(b) a superposition of several photos taken during the experiment is shown, clearly indicating the amplitude of the plant's left-right motion.

¹ <https://www.raspberrypi.org/products/camera-module/>

² Find a video at: <https://youtu.be/e-84bxhwpZo>

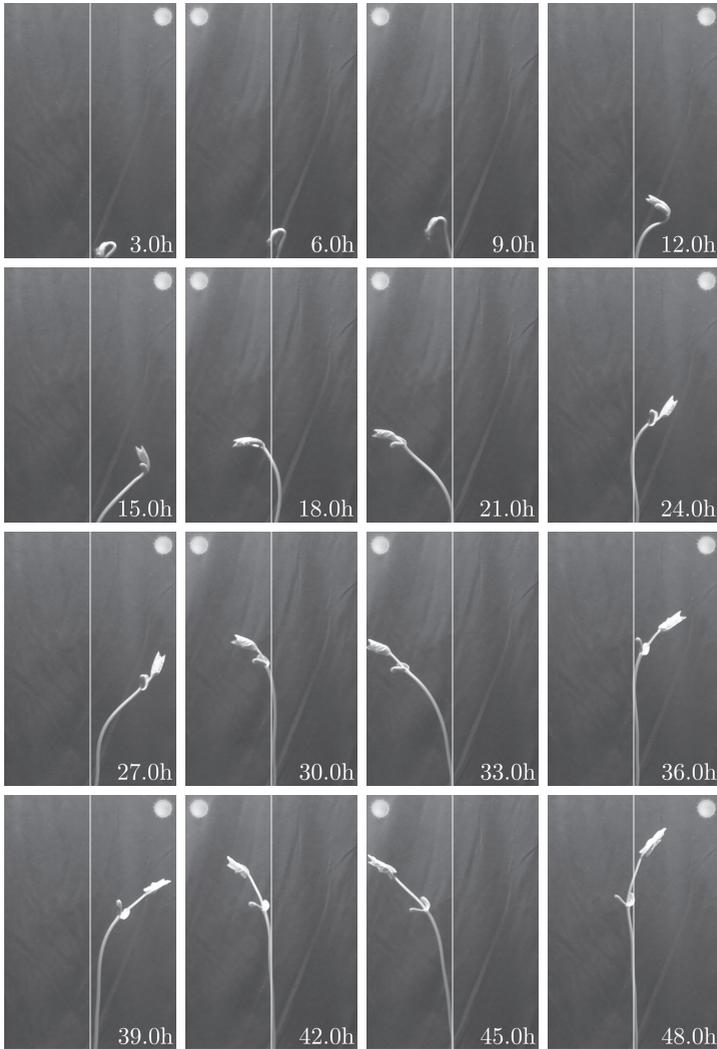


Figure 2: Photos of the bean plant at different times during the experiment showing rapid growth and motion towards light; the white line indicates the location of the roots, the sun symbolizes the activation of a light source to the left or right.

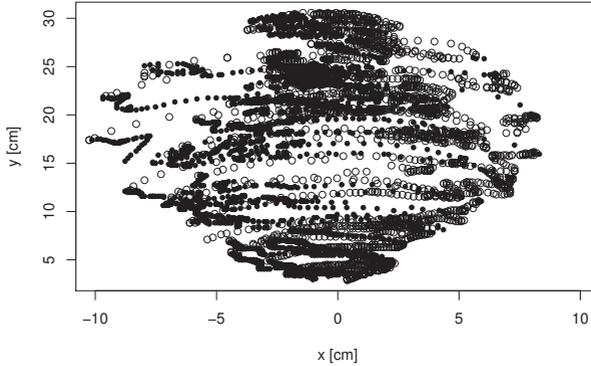


Figure 3: Plant tip positions from all our experiments. Black filled circles denote the activation of the left, white filled circles the activation of the right light source.

Bean growth and motion model

We processed the obtained 2D images using the OpenCV library. Due to the high contrast between background and plant, we simply transformed the images to grayscale, applied a Gaussian filter to smoothen the images and then extracted the brightest points. The highest point of the plant (after cropping the area of actual plant growth, dismissing the pot and the light sources) is stored as its tip with the position $\mathbf{x} = (x, y)$ relative to the roots of the plant. The time series \mathbf{x}_t for each experiment is a rough description of the plant's growth process and the effect of the controlled stimuli (Fig. 3).

We make use of this data to create a simple *bean growth–motion model*. We define a model that represents a plant's current tip position \mathbf{x}_t and the current lighting condition L_t (boolean value indicating whether the left light is on). A current configuration of the system is then defined by $(\mathbf{x}, L)_t$. Using the collected data we derive the next tip position \mathbf{x}_{t+1} for discrete time steps, each representing five minutes of real time. We assume that the two light sources are identical and also that the plant has no other bias to grow towards either of the two directions. Therefore, we mirrored the collected data to both sides (mapping $x \mapsto -x$ and keeping y identical) to

logically double the available data and hence to increase the precision of the model (note that the light conditions can still be distinguished: light on in the same quadrant or light on in the other quadrant).

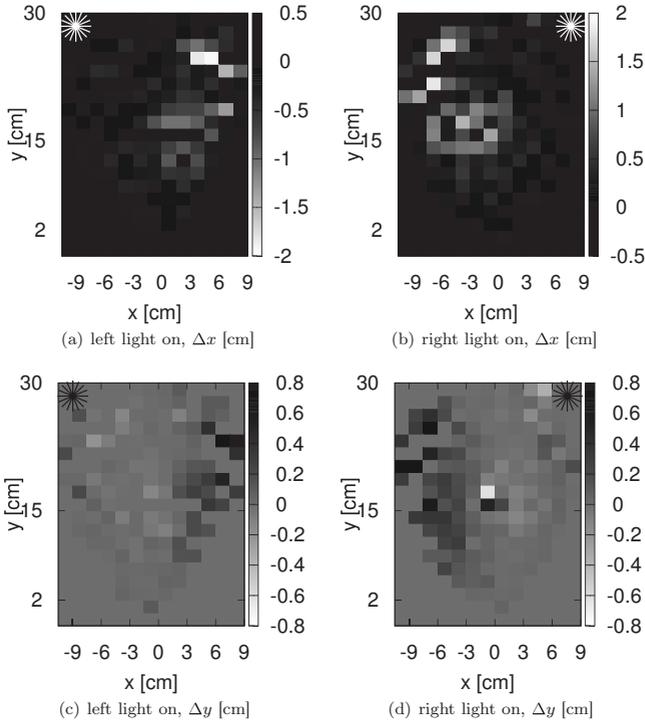


Figure 4: xy -motion of the bean plant's tip within 5min when activating the left/right light source (left-hand/right-hand column, respectively).

In order to calculate the next plant tip position \mathbf{x}_{t+1} for a given configuration $(\mathbf{x}, L)_t$, we define a rectangle $R = ((x - w_x, y - w_y), (x + w_x, y + w_y))$ with the plant tip position \mathbf{x} at the center, width $2w_x$, and height $2w_y$ (i.e., a sliding window). Then we select all data points that are contained by rectangle R and that have the same light condition. Based on these data points we want to calculate a shift of the tip position Δx and Δy . From the selected data points and their successors in time we obtain samples of

Δx and Δy . In Fig. 4 we show the original data of Δx and Δy within 5min time periods for all data points and both light settings before mirroring the data. The plant tip shift at the side opposite to the currently active light has the greatest absolute values. Note that, not only Δx but also Δy are subject to the plant’s rotational motion which is usually centered at $x = 0$. We observe that at the bottom half ($y < 15$) the change in motion is generally smaller, probably because the plant’s size limits its motion. In Fig. 5 we show the distribution of Δx and Δy for all data points. We interpret these distributions as normal distributions which can be expected for such a natural growth process.

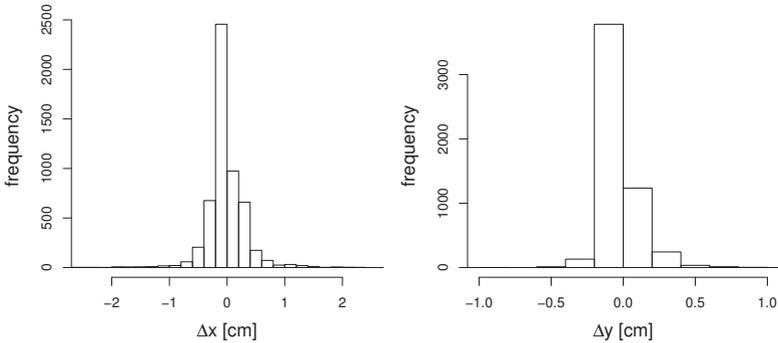


Figure 5: Histograms of plant tip position shifts acquired from all data points within 5min time period.

We present and investigate three different methods to calculate the tip shift. The first is called *deterministic*, implements a deterministic model, and uses the mean values $\overline{\Delta x}$ and $\overline{\Delta y}$ of all data points inside the rectangular window. The second is called *stochastic*, implements a a stochastic model, and directly samples uniformly from the data points inside the rectangular window. The third method is called *mixed*, implements a mixture of the previous two methods, where the mean value $\overline{\Delta x}$ is used to calculate Δx , while Δy is randomly sampled from a normal distribution with a mean value of 0.04 and a standard deviation of value 0.01. The mean value of 0.04 was chosen to get overall heights of about 15cm for the chosen experiment length which corresponds to what we observed for the experiments with the natural plant.

Table 1: Used NEAT parameters.

Parameter	Value	Parameter	Value
<i>PopulationSize</i>	50	<i>CrossoverRate</i>	0.5
<i>DynamicCompatibility</i>	True	<i>MutateWeightsProb</i>	0.9
<i>YoungAgeTreshold</i>	15	<i>YoungAgeFitnessBoost</i>	1.0
<i>OverallMutationRate</i>	0.5	<i>WeightReplacementMax</i>	5.0
<i>MinSpecies</i>	5	<i>WeightMutationRate</i>	0.75
<i>MaxSpecies</i>	25	<i>Elitism</i>	0.1
<i>SurvivalRate</i>	0.6	<i>MutateAddNeuronProb</i>	0.04

Evolutionary approach

We use MultiNEAT [2] in combination with our simple growth–motion model to evolve closed-loop controllers in simulation. MultiNEAT is a portable software library implementing NEAT (NeuroEvolution of Augmenting Topologies) that uses the complexification method [12] to evolve artificial neural networks (ANN). Table 1 specifies the NEAT parameters used in our experiments. These parameters are based on our experience from previous experiments where we evolved robot controllers for a parallel parking task [16]. We evolve ANN with two input neurons, a variable number of neurons in the hidden layer (determined by NEAT), one output neuron, and an unsigned step activation function.

The input of the network is the current plant tip position $\mathbf{x}_t = (x, y)$ (initially $\mathbf{x}_0 = (0, 0)$). For each time step (discrete time steps represent 5min in reality), the network’s output simply determines the light condition L_t , that is, whether the left light or the right light is turned on. Next, the current configuration of the system $(\mathbf{x}, L)_t$ is passed to the simple growth–motion model in order to obtain the next tip position \mathbf{x}_{t+1} . This procedure is iterated for 300 time steps (representing 25 hours of real time), while evaluating the performance of the controller in respect to a certain task.

As a proof of concept, we evolve controllers for two simple tasks. (1) *maximum motion*: The task is to maximize the plant’s overall motion (i.e., covered distance of the plant tip) during a time period of 25 hours, and (2) *three targets*: The plant has to approach three different desired positions

in space during the experiment. For both tasks, we conduct a set of three experiments using the above mentioned *deterministic*, *mixed* and *stochastic* method (i.e., calculation of the next plant tip position \mathbf{x}_{t+1}) to model the growth and motion behavior of the plant.

First, we define the fitness function F (eq. 1) to evaluate the performance of the individuals during the evolutionary processes within the maximum motion experiments. It simply accumulates the absolute value of all the plant tip position shifts Δx_t over 300 time steps:

$$F = \sum_{t=2}^{300} |\Delta x_t|, \text{ with } \Delta x_t = x_t - x_{t-1}. \quad (1)$$

For each experiment, 10 evolutionary runs of 100 generations each with a population size of 50 were performed. For the *deterministic* and *mixed* method we use only one repetition to evaluate a controller. For the *stochastic* method we do three repetitions, hence we get three fitness values F_0, F_1, F_2 , and we define the controller's fitness conservatively as the minimum $F = \min\{F_0, F_1, F_2\}$. The best fitness of each run over generations for the *deterministic* method in the *maximum motion* experiment is shown in Fig. 6(a). Clearly, the task here is simple, therefore, convergence is achieved already at about generation 60. The behavior of the simulated plant tip (i.e., its trajectory) when running the best controller is shown in Fig. 7(a). The controller turns the right light on in the beginning, keeps it on to grow the plant tip to the right, then switches the left light on, and switches between the lights during the last few time steps. The fitness value of this controller is $F = 72.7$, which means the tip moved 72.7cm in total (horizontally). The best fitness for each run over generations for the *mixed* method for the *maximum motion* experiment is shown in Fig. 6(b). Similarly, the evolutionary approach converges quickly because of the task's simplicity. In Fig. 7(b) we show a trajectory of the plant tip sampled from the stochastic simulation (plant tip motion in height is stochastic, motion in width is deterministic) when running the best controller with a fitness value $F = 135.5$. The controller is turning lights on and off frequently to maximize the plant motion. The best fitness over generations for the *stochastic* method in the *maximum motion* experiment is shown in Fig. 6(c). Here, the complexity of the task is increased because both the plant tip's

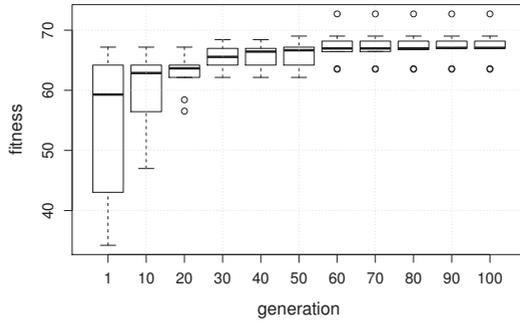
motion in height and width are stochastic and also variant because they are sampled directly now. As a result, we observe a non-monotonic increase of fitness interrupted by sudden drops. There is only a small increase in fitness within 100 generations and no saturation. We have tested the initial populations against the final populations using the Wilcoxon Rank-Sum Test to check whether the evolved controllers perform better than random. The test indicates that the controllers of generation 100 are significantly better. The simulated behavior of the tip when running the best controller of fitness value $F = 96.57$ is shown in Fig. 7(c).

Considering the objective to reach *three targets*, the fitness function is defined as

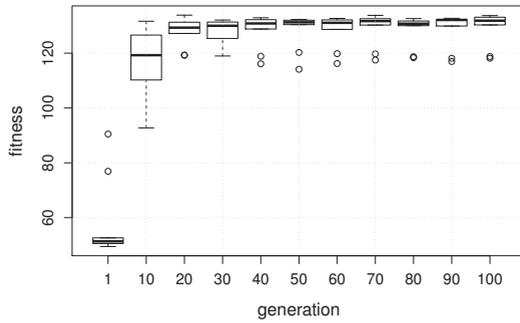
$$\begin{aligned}
 D(\mathbf{x}_{\text{target}}) &= |x_{\text{target}} - x_t| + |y_{\text{target}} - y_t|, \\
 E &= \sum_{t=2}^{120} D(3, 8) + \sum_{t=121}^{220} D(-5, 11) + \sum_{t=221}^{300} D(-1, 18), \\
 F &= F_{\text{max}} - E,
 \end{aligned} \tag{2}$$

for an assumed maximal fitness $F_{\text{max}} = 3000$. For each target it considers a defined time interval during which it accumulates the differences between the respective target's position $\mathbf{x}_{\text{target}}$ and the current plant tip position \mathbf{x}_t . The first phase lasts for 120 time steps (time interval $[0, 120]$), the second for 100 time steps (time interval $[120, 220]$), and the third phase for 80 time steps (time interval $[220, 300]$).

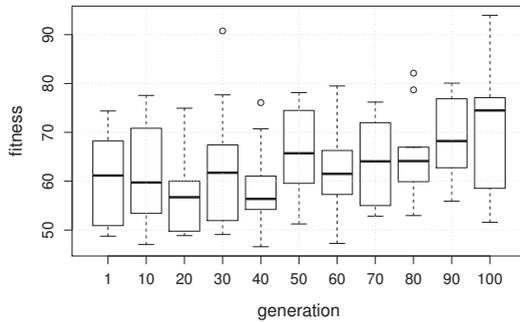
Here, for both the *deterministic* and *mixed* experiments, 10 evolutionary runs of 600 generations each with a population size of 50 were performed. Similarly, we also conducted the stochastic experiment, however, for only 150 generations (due to technical issues) but still with three repetitions and the minimum of these three values as fitness. In contrast to the maximum motion task, this task is more complex and requires longer evolutionary runs in order to evolve successful behaviors. The best fitness of each run over generations for the *deterministic* method in the *three targets* experiment is shown in Fig. 8(a). In contrast to the results from the *maximum motion* experiment, convergence is not achieved. Instead, there is a steady increase in performance through the 600 generations. This proves that the task here is more complex and indicates the presence of a room for improvement. The behavior of the simulated plant tip when



(a) deterministic

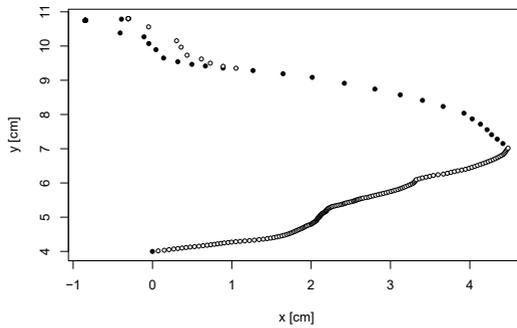


(b) mixed

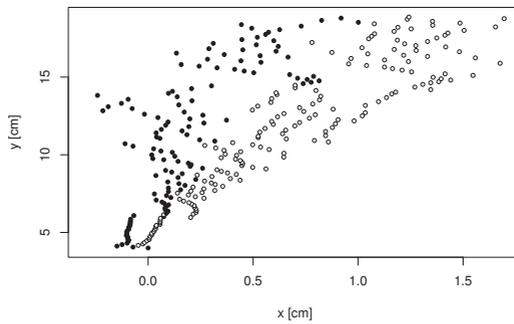


(c) stochastic

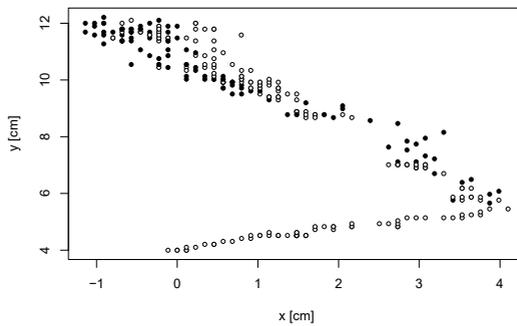
Figure 6: Maximum motion, fitness of the best controllers per generation of the 10 evolutionary runs for the maximum motion experiments.



(a) deterministic



(b) mixed



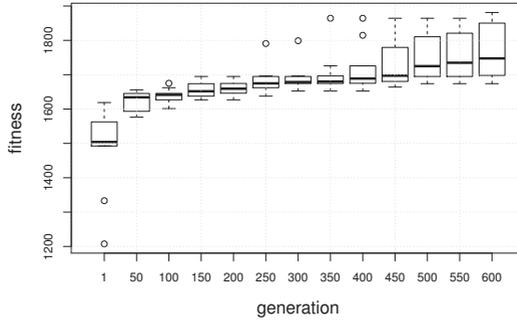
(c) stochastic

Figure 7: Maximum motion, (sampled) trajectories of the simulated plant tip (white filled circle: right light is on, black filled circle: left light is on) for the respective best controller.

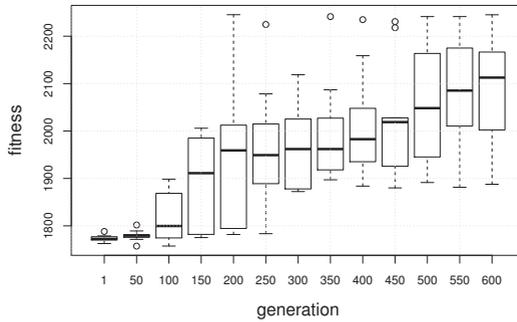
running the best controller is shown in Fig. 9(a). The controller turns the right light on in the beginning until the plant tip is close to the first target, then switches the left light on until the plant tip is close to the left target, then keeps on switching between the lights and fails to meet the third target. The fitness value of this controller is $F = 1881.2$. The best fitness of each run over generations for the *mixed* method in the *three targets* experiment is shown in Fig. 8(b). Similarly, no saturation is observed, however, the trajectory of the plant tip when running the best controller with a fitness value $F = 2255.9$ (see Fig. 9(b)) shows that the tip could successfully approach all three targets. The best fitness over generations for the *stochastic* method in the *three targets* experiment is shown in Fig. 8(c). As above, we have checked for significance between the initial and final population. According to the Wilcoxon Rank-Sum Test, the performance of controllers in the last generation are not significantly better than the initial population. However, according to our observations (see Fig. 9(c)), there is an improvement in the performance and longer evolutionary runs are planned for future work. The simulated behavior of the tip when running the best controller of fitness value $F = 1843.65$ is shown in Fig. 9(c).

Conclusion and future work

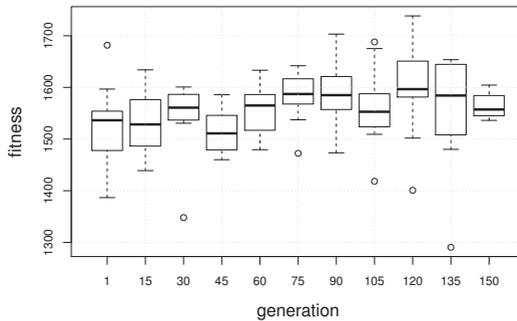
We have reported on experiments with a natural plant, the common bean (*Phaseolus vulgaris*). We have introduced a simple plant growth and motion model which focuses on the plant tip exclusively. Then we used this model to simulate the plant tip motion in several preliminary experiments. This way we evolve closed-loop controllers that maximize the plant's motion or grow the tip to three differently located targets. According to the discussion in the previous section, the results from the six experiments indicate the effectiveness of our evolutionary approach (arguably except for the experiments with the *stochastic* method). Also, in many cases the optimization process did not fully converge. Running the evolutionary process for a longer time would probably increase the performance of the evolved controllers. Therefore, we intend to investigate longer evolutionary runs in future work, probably along with a more efficient implementation.



(a) deterministic

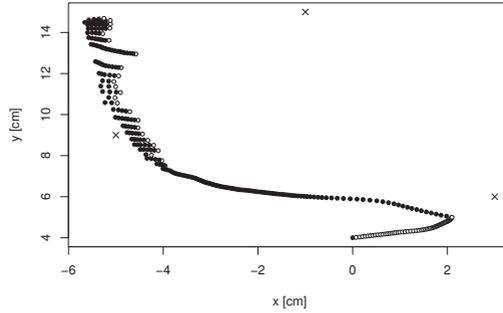


(b) mixed

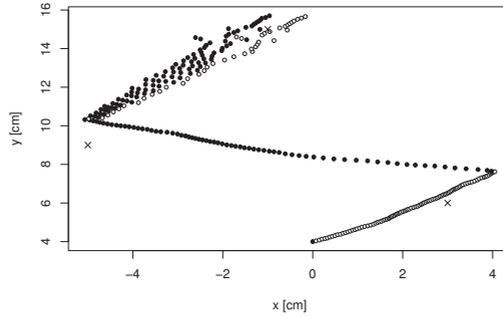


(c) stochastic

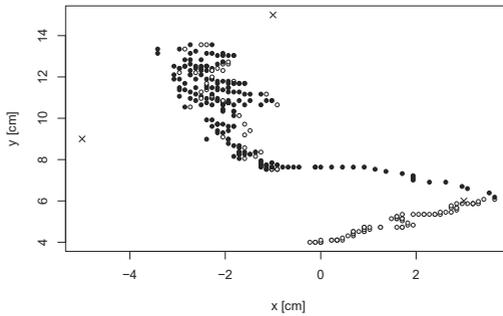
Figure 8: Three targets experiments, fitness of the best controllers per generation of the 10 evolutionary runs for the three targets experiments.



(a) deterministic



(b) mixed



(c) stochastic

Figure 9: Three targets experiments, (sampled) trajectories of the simulated plant tip (white filled circle: right light is on, black filled circle: left light is on) for the best controllers. Black crosses indicate the positions of the three targets.

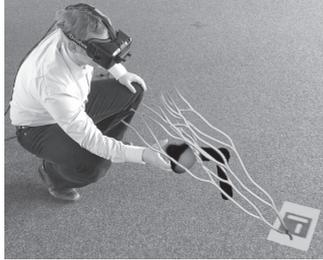


Figure 10: This photomontage illustrates the workings of our current augmented reality interface for bio-hybrid plant-robot systems.

We are also motivated to investigate the impact of different parameters of the evolutionary algorithm (e.g., the mutation rate) in order to decrease the time required to evolve successful behaviors.

Especially the results from the mixed, *three targets* experiment are promising (see Fig. 9(c)). We plan to test these controllers, that were evolved in simulation, in the real setting and hence investigate the reality gap and the reaction of real plants. As the controllers will not perfectly transfer into reality, we expect to require additional investigations. We plan to extend the existing model of the plant tip to consider the whole plant growth and motion behavior, that is, also to model lower parts of the plant.

In addition to honing the domain model and the developmental and dynamic plant representation, we are also pushing towards the application of *flora robotica*'s results (e.g., applications in architecture). In terms of accessibility, we found it especially useful to spatially experience a bio-hybrid plant-robot system. To this end, we are developing an augmented reality interface that allows the user to project the system into arbitrary real world spaces, configure and fast forward the system's evolution. Fig. 10 shows our current prototype: (1) A stereoscopic camera (OVRVision) is mounted on a stereoscopic head-mounted display (Oculus DK2). (2) QR code markers on the ground synchronize the augmented reality projection with the lab space. (3) The user can place robotic machinery, seed plants and experience their interplay over time.

Finally, we will integrate these approaches within our project *flora robotica* to control the growth and motion of natural plants by robots and to create an adaptive bio-hybrid system.

Acknowledgment

Project ‘*flora robotica*’ has received funding from the European Union’s Horizon 2020 research and innovation program under the FET grant agreement, no. 640959.

References

- [1] J. C. Bongard. Evolutionary robotics. *Communications of the ACM*, 56(8):74–83, 2013.
- [2] P. Chervenski and S. Ryan. MultiNEAT, project website;. <http://www.multineat.com/>.
- [3] N. Correll, N. Arechiga, A. Bolger, M. Bollini, B. Charrow, A. Clayton, F. Dominguez, K. Donahue, S. Dyar, L. Johnson, et al. Indoor robot gardening: design and implementation. *Intelligent Service Robotics*, 3(4):219–232, 2010.
- [4] A. E. Eiben and J. Smith. From evolutionary computation to the evolution of things. *Nature*, 521:476–482, May 2015.
- [5] R. Featherstone and D. Orin. Robot dynamics: Equations and algorithms. In *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on*, volume 1, pages 826–834. IEEE, 2000.
- [6] *flora robotica*. project website, 2015. <http://www.florarobotica.eu>.
- [7] P. C. Garzón and F. Keijzer. Plants: Adaptive behavior, root-brains, and minimal cognition. *Adaptive Behavior*, 19(3):155–171, 2011.
- [8] H. Hamann, M. Wahby, T. Schmickl, P. Zahadat, D. Hofstadler, K. Stoy, S. Risi, A. Faina, F. Veenstra, S. Kernbach, I. Kuksin, O. Kernbach, P. Ayres, and P. Wojtaszek. *flora robotica* – mixed societies of symbiotic robot-plant bio-hybrids. *Proceedings of the 2015 IEEE Symposium on Artificial Life (IEEE ALIFE’15)*, 2015. (in press).

- [9] S. Koos, J.-B. Mouret, and S. Doncieux. The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation*, 17(1):122–145, 2013.
- [10] A. Lindenmayer. Developmental algorithms for multicellular organisms: A survey of L-systems. *Journal of Theoretical Biology*, 54(1):3–22, 1975.
- [11] D. Slaughter, D. Giles, and D. Downey. Autonomous robotic weed control systems: A review. *Computers and electronics in agriculture*, 61(1):63–78, 2008.
- [12] K. O. Stanley and R. Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21(1):63–100, Jan. 2004.
- [13] O. Stava, S. Pirk, J. Kratt, B. Chen, R. Měch, O. Deussen, and B. Benes. Inverse procedural modelling of trees. In *Computer Graphics Forum*, volume 33, pages 118–131. Wiley Online Library, 2014.
- [14] D. W. Thompson. *On Growth and Form*. Cambridge University Press, 1917.
- [15] S. Von Mammen and C. Jacob. The evolution of swarm grammars-growing trees, crafting art, and bottom-up design. *IEEE Computational Intelligence Magazine*, 4(3):10–19, 2009.
- [16] M. Wahby and H. Hamann. On the tradeoff between hardware protection and optimization success: A case study in onboard evolutionary robotics for autonomous parallel parking. In *Applications of Evolutionary Computation (EvoApplications 2015)*, volume 9028 of *LNCS*, pages 759–770. Springer, 2015.
- [17] R. A. Watson, S. G. Ficici, and J. B. Pollack. Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18, 2002.
- [18] M. Woolridge. *Introduction to Multiagent Systems*. John Wiley & Sons, Inc. New York, NY, USA, 2001.
- [19] A. Zamuda and J. Brest. Vectorized procedural models for animated trees reconstruction using differential evolution. *Information Sciences*, 278:1–21, 2014.

SACOBRA: Self-Adjusting Constrained Black-Box Optimization with RBF

Samineh Bagheri¹, Wolfgang Konen¹, Christophe Foussette²,
Peter Krause², Thomas Bäck³, Patrick Koch¹

¹TH Köln (University of Applied Sciences)

Steinmüllerallee 1

51643 Gummersbach

E-Mail: {wolfgang.konen, samineh.bagheri}@th-koeln.de

²divis intelligent solutions GmbH

Joseph-von-Fraunhofer-Str. 20

44227 Dortmund

E-Mail: {foussette, krause}@divis-gmbh.de

³Leiden University, LIACS

2333 CA Leiden, The Netherlands

E-Mail: T.H.W.Baeck@liacs.leidenuniv.nl

Abstract

Modern real-world optimization problems are often high dimensional and subject to many constraints. These problems are typically expensive in terms of cost and computational time. Conventional constraint-based solvers often require a high number of function evaluations which are not affordable for such problems in practice. Employment of fast surrogate models to approximate objective and constraint functions is a known approach for efficient optimization. In this paper we present a new algorithm called **Self-Adjusting COBRA (SACOBRA)** based on Regis' COBRA [11]. We evaluate our approach by using 11 G-problems. We get very good results on 10 of the 11 G-problems with a severely limited budget of only 300 evaluations.

Introduction

Constrained optimization is often much harder than unconstrained optimization since the multitude of functions (one objective, numerous constraints) may constitute highly conflicting goals. If these functions are in addition expensive to evaluate, most algorithms make use of surrogate models.

The strength of surrogate-assisted techniques relies on the correct choice of modeling technique. In contrast to many surrogate modeling techniques like Kriging and support vector machines [8, 3], the performance of **radial basis function (RBF)** is only weakly dependent on the dimensionality of the optimization tasks or number of design points [2]. Therefore, in the area of efficient surrogate-assisted optimization a lot of attention is devoted to RBF modeling [11, 12, 4, 5]. COBRA [11] – an efficient constrained-based optimizer which uses RBF surrogates – outperforms many other algorithms on a large number of benchmarks. COBRA-R [4, 5] is a variant of COBRA implemented in R [10] with extended initialization methods and a novel repair technique for infeasible solutions [5].

Although both extensions gained remarkable success, the achievement was reached only after careful selection of the parameters and preprocessing transformations for each problem [4]. In the case of black-box optimization manual parameter tuning is not a valid approach since it requires some information about each single problem. Furthermore, automatic exploration of the parameter space is expensive. Therefore, some recent studies focused on automatic parameter tuning based on some measurable features of the problem [1].

In this extended abstract, we outline the goal to develop *one* algorithm which decides on its own about the best parameter setting and transformation functions. The new algorithm, which we call **Self-Adjusting COBRA (SACOBRA)**, is able to adjust those parameters automatically after the initialization phase. Additionally, SACOBRA incorporates a **random restart** algorithm to avoid occasional early stagnation and bad solutions due to an unfortunate initial design.

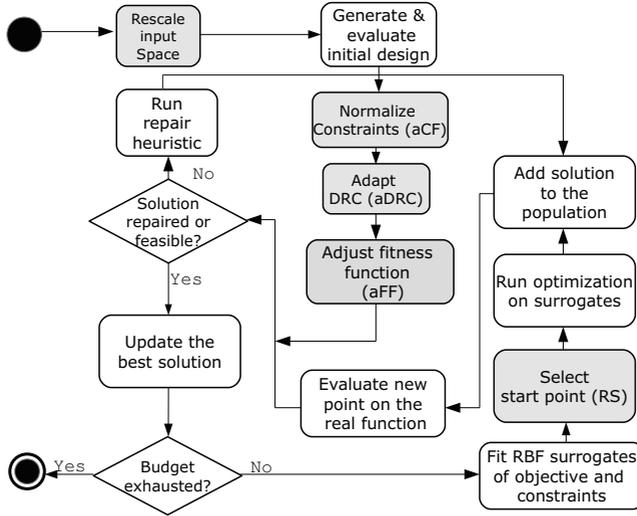


Figure 1: SAOCBRA algorithm proposed in this paper. The SACOBRA extensions are the grey boxes. Using only the white boxes results in the former COBRA-R algorithm.

Methods

Constrained optimization

A constrained optimization problem can be defined by the minimization of an objective function f subject to constraint function(s) g_1, \dots, g_m :

$$\begin{aligned} &\text{Minimize} && f(\vec{x}), \vec{x} \in \mathbb{R}^d \\ &\text{subject to} && \\ &&& g_i(\vec{x}) \leq 0, i = 1, 2, \dots, m \end{aligned}$$

In this paper we always consider minimization problems. Maximization problems are transformed to minimization without loss of generality.

COBRA: Constrained Optimization by Radial Basis Function Approximation

Constrained Optimization by Radial Basis Function Approximation (COBRA) (Figure 1, only white boxes) is an efficient optimization algorithm proposed by Regis [11]. This method relies on modeling the objective function and constraint function(s) by means of radial basis functions (surrogates) in order to minimize evaluations of the real objective and constraint functions. Each iterate is a result of an optimization on the surrogates. Only at the end of each iteration the real functions are evaluated once and a new solution is added to the population. COBRA-R [4] is a re-implementation of COBRA [11] in R [10] with some extensions which are described in detail in [4].

An important element of COBRA and COBRA-R is the *Distance Requirement Cycle (DRC)*: In order to facilitate exploration, Regis [11] proposes to add in each iteration i additional constraints enforcing that the current solution \vec{x} is at least a distance ρ_i away from all previous iterates. The distance ρ_i is taken cyclically from a small set of distances, the so-called DRC. Large ρ_i enforce exploration, small ρ_i exploitation. It turns out that different problems are quite sensitive to the right choice of DRC.

Repair algorithm RI-2

Sometimes the infill points returned by the internal optimizer are infeasible. A repair algorithm is embedded in the COBRA-R optimization framework which intends to repair infill points with a slight infeasibility by guiding them to the feasible region. The repair algorithm RI-2 used in COBRA-R is described and discussed in detail in [5]. It is worthwhile to mention that the repair algorithm is performed on the surrogate models, so no real function evaluations are necessary for this repair.

SACOBRA: Self-Adjusting COBRA

SACOBRA, the self-adjusting COBRA-R algorithm, includes five extensions (Figure 1, with grey boxes) in comparison to the COBRA-R optimization framework [5]:

- **Rescale the input space:** The input vector \vec{x} is element-wise rescaled to $[-1, +1]$. This helps to have a better exploration all over the search space because all dimensions are treated the same and also avoids numerical instabilities caused by high values of \vec{x} .
- **Random Start algorithm (RS):** Normally COBRA starts optimization from the best point found so far. With RS the optimization starts with a certain probability from a random point in the search space. RS is especially beneficial when the search gets stuck in local optima.
- **Adjusting DRC parameter (aDRC)** is done after the initialization phase. Our experimental analysis showed that large DRC values set can be harmful for problems with a very steep objective function, while those large values are needed for other problems. Therefore, we developed an automatic DRC adjustment which selects the appropriate DRC set according to the information extracted after the initialization phase.
- **Adjusting fitness function (aFF)** is developed to modify objective functions which are challenging to model with RBF. Our experimental analysis revealed that RBF often do not provide a proper model for functions which are very steep. A logarithmic transformation of the objective function can be helpful. SACOBRA internally decides whether a logarithmic transformation of the objective function is necessary.
- **Adjusting constraint function(s) (aCF)** is actually done by normalizing the range of constraint functions for each problem. Embedding this step boosts up the optimization performance because all constraints have values in a similar range.

The details of the internal adjustment algorithms and the role of problem-specific features like *FR* and *GR* (see Tab. 1 below) will be presented in a forthcoming publication.

Experimental Setup

We evaluate our proposed technique by using a well-studied test suite of G-problems described in [6]. The diversity of the G-problem characteristics makes them a very challenging benchmark for optimization techniques. In Table 1 we show features of these functions. In this study we identified two new features *FR* and *GR* (defined in Table 1) which constitute useful elements for our self-adjusting procedures.

We apply the SACOBRA algorithm to 330 problems: G01-G11, initialized with 30 different initial populations. The initial population of size $3 \cdot d$ is generated by means of Latin hypercube sampling. The optimization on surrogates is done by COBYLA [9]. Cubic RBFs are utilized to model the functions. The DRC parameters are selected internally among $DRCL = \langle 0.3, 0.05, 0.001, 0.0005, 0.0 \rangle$ and $DRCs = \langle 0.001, 0.0 \rangle$. In order to measure the strength of each extension we repeat all tests in absence of each extension. The importance of each extension is measured with the Wilcoxon signed rank sum test of the final optimization error (Table 2), which is a one sided, paired test on the 30 pairs with different seeds.

Table 1: Characteristics of the G-functions: d : dimension, ρ^* : feasibility rate (%) after changing equality constraints to inequality constraints, *FR*: range of the fitness values, *GR*: ratio of largest to smallest constraint range, LI: number of linear inequalities, NI: number of nonlinear inequalities, NE: number of nonlinear equalities, a : number of constraints active at the optimum.

Fct.	d	type	ρ^*	<i>FR</i>	<i>GR</i>	LI	NI	NE	a
G01	13	quadratic	0.0003%	298.14	1.969	9	0	0	6
G02	10	nonlinear	99.997%	0.57	2.632	1	1	0	1
G03	20	nonlinear	0.0000%	92684985979.23	1.000	0	0	1	1
G04	5	quadratic	26.9217%	9832.45	2.161	0	6	0	2
G05	4	nonlinear	0.0919%	8863.69	1788.74	2	0	3	3
G06	2	nonlinear	0.0072%	1246828.23	1.010	0	2	0	2
G07	10	quadratic	0.0000%	5928.19	12.671	3	5	0	6
G08	2	nonlinear	0.8751%	1821.61	2.393	0	2	0	0
G09	7	nonlinear	0.5207%	10013016.18	25.05	0	4	0	2
G10	8	linear	0.0008%	27610.89	3842702	3	3	0	3
G11	2	linear	66.7240%	4.99	1.000	0	0	1	1

Table 2: Wilcoxon rank sum test, paired, one sided, significance level 5%. Shown is the p-value. Significant tests ($p < 5\%$) are marked in boldface. Optimization methods: M1: SACOBRA, M2: SACOBRA\rescale (SACOBRA without rescaling the input space), M3: SACOBRA\RS (SACOBRA without random start), M4: SACOBRA\adRC, M5: SACOBRA\aff, M6: SACOBRA\acf, M7: COBRA ($X_i = DRCS$), M8: COBRA ($X_i = DRCL$).

func\method	M1M2	M1M3	M1M4	M1M5	M1M6	M1M7	M1M8
G01	0.177	4.8e-02	0.887	1.000	1.000	0.167	4.8e-02
G02-10d	0.144	0.664	0.743	0.789	0.757	0.074	0.985
G03	1.000	8.7e-03	1.000	1.4e-06	1.000	2.4e-05	1.4e-06
G04	0.230	0.121	1.000	0.909	0.909	0.121	9.1e-07
G05	1.4e-03	0.557	1.000	0.676	0.138	0.999	0.997
G06	9.1e-07	0.745	1.000	0.230	0.500	0.194	9.1e-07
G07	0.745	0.516	1.000	0.352	0.352	0.516	2.0e-06
G08	7.6e-03	0.112	3.7e-03	1.000	1.000	5.4e-04	0.112
G09	0.971	0.594	1.000	1.5e-05	0.884	1.5e-05	1.4e-06
G10	3.1e-03	2.7e-03	1.000	0.843	2.5e-02	1.1e-03	2.0e-03
G11	0.660	1.000	0.875	1.000	1.000	1.000	1.000

Results and Discussion

In our previous work [4] we established good results with COBRA-R on most G-problems, but we had to manually tune algorithmic configurations and parameters for each G-problem anew, which is a tedious and time-consuming procedure. Now we are in a position to present with SACOBRA an algorithm which runs with the same settings on all G-problems and produces good results with relatively few function evaluations (100–300). We will show in the talk and in a forthcoming publication the performance profiles [7] for SACOBRA and its variants, leading to a substantial improvement over COBRA-R with *fixed* parameter settings. SACOBRA solves 85% of the 330 test problems while COBRA-R solves only 60%.

We can identify with our results the elements of the SACOBRA algorithm with greatest impact: Most important are **rescaling** and **random restart**, followed by **automatic fitness function adjustment**. Least important are **adRC** and **acf**.

Tab. 2 shows that each of these elements has its importance (is relevant) for some of the G-problems: M1 is significantly better than M* at least for some G-problems. And each G-problem benefits from one or more SACOBRA extensions. The only exceptions from this rule are G02-10d and G11, but for different reasons: G02-10d is a high-dimensional and highly multimodal problem which is generally hard to solve by *any* of the SACOBRA- or COBRA-variants (the reason is that surrogate models with a low number of points cannot capture enough detail of this complicated fitness function). G11 on the other hand is an easy problem which is solved by *all* SACOBRA variants, so none is significantly better than the others.

Conclusion

We presented with SACOBRA a self-adjusting algorithm for expensive constrained optimization which can successfully solve a variety of challenging benchmark problems without any problem-specific parameter tuning. It does so by self-adjusting its internal parameters to the characteristics of the problem, either after the initialization phase or online during iterations.

SACOBRA solves 10 of the 11 G-problems. The only exception is G02-10d, which is a highly multimodal problem. Such problems cannot be solved in a few iterations with the current surrogate models. The investigation of surrogate modeling for highly multimodal functions is a topic of our future research.

Acknowledgements

This work has been supported by the Bundesministerium für Wirtschaft (BMW) under the ZIM grant MONREP (AiF FKZ KF3145102, Zentrales Innovationsprogramm Mittelstand).

Supported by:



Federal Ministry
for Economic Affairs
and Energy

on the basis of a decision
by the German Bundestag

References

- [1] Jakob Bossek, Bernd Bischl, Tobias Wagner, and Günter Rudolph. Learning feature-parameter mappings for parameter tuning via the profile expected improvement. In *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference, GECCO '15*, pages 1319–1326, New York, NY, USA, 2015. ACM.
- [2] Martin D. Buhmann. *Radial Basis Functions*. Cambridge University Press, New York, NY, USA, 2003.
- [3] Michael Emmerich, Alexios Giotis, Mutlu Özdemir, Thomas Bäck, and Kyriakos Giannakoglou. Metamodel-assisted evolution strategies. In Juan Julián Merelo Guervós et al., editors, *Proceedings of the Conference on Parallel Problem Solving from Nature (PPSN VII)*, volume 2439 of *Lecture Notes in Computer Science*, pages 361–370. Springer Berlin Heidelberg, 2002.
- [4] Patrick Koch, Samineh Bagheri, Christophe Foussette, Peter Krause, Thomas Bäck, and Wolfgang Konen. Constrained optimization with a limited number of function evaluations. In Frank Hoffmann and Eyke Hüllermeier, editors, *Proceedings of the 24th Workshop on Computational Intelligence*, pages 119–134. Universitätsverlag Karlsruhe, 2014. Young Author Award GMA-CI.
- [5] Patrick Koch, Samineh Bagheri, Wolfgang Konen, Christophe Foussette, Peter Krause, and Thomas Bäck. A new repair method for constrained optimization. In Juan Luis Jiménez-Laredo, editor, *Proceedings of the Genetic and Evolutionary Computation Conference 2015 (GECCO)*, pages 273–280. ACM, 2015.
- [6] Z. Michalewicz and M. Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1):1–32, 1996.
- [7] Jorge J. Moré and Stefan M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM J. Optimization*, 20(1):172–191, 2009.
- [8] J. Poloczek and O. Kramer. Local svm constraint surrogate models for self-adaptive evolution strategies. In *Proceedings of the 36th Annual German Conference on Advances in Artificial Intelligence (KI 2013)*, pages 164–175. Springer, 2013.
- [9] M.J.D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances In Optimization And Numerical Analysis*, pages 51–67. Springer, 1994.

- [10] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.
- [11] R.G. Regis. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Engineering Optimization*, 46(2):218–243, 2013.
- [12] Chaoli Sun, Yaochu Jin, Jianchao Zeng, and Yang Yu. A two-layer surrogate-assisted particle swarm optimization algorithm. *Soft Computing*, 19(6):1461–1475, 2015.

Locally weighted regression through data imprecisiation

Shenzhen Lu, Eyke Hüllermeier

Department of Computer Science
University of Paderborn

E-Mail: lushenzhen@gmail.com, eyke@upb.de

Introduction

This extended abstract outlines the idea of realizing locally weighted learning and statistical inference within the framework of *superset learning*. More specifically, we propose an alternative to standard locally weighted linear regression, which is commonly used in statistics and machine learning. Our approach is based on replacing precisely observed output values by intervals—a process we refer to as “data imprecisiation”. As will be explained in more detail later on, the influence of an observation can thus be controlled by the length of the corresponding interval.

Our approach builds on a generic framework for superset learning that we recently introduced in [5, 6], and that will be briefly recalled in the next section. The main purpose of this framework is to support the systematic development of methods for learning from imprecise or ambiguous data, namely, training data that is characterised in terms of sets of candidate values. Additionally, however, it can be used for learning from standard (precise) data, which is deliberately turned into imprecise data. In this way, different effects can be achieved, including the one already mentioned, namely the weighing of the influence of a training example on the overall result of the learning process: the more imprecise an observation is made, the less it will influence the model or prediction induced from the data.

In the next section, we recall our generic approach to superset learning based on generalized loss minimization. Then, we show how this approach can be used to develop an alternative method for locally weighted linear regression. Prior to concluding, we present some experimental results.

Superset Learning

Superset learning is a specific type of learning from weak supervision, in which the outcome (response) associated with a training instance is only characterized in terms of a subset of possible candidates. Thus, superset learning is somehow in-between supervised and semi-supervised learning, with the latter being a special case (in which supersets are singletons for the labeled examples and cover the entire output space for the unlabeled ones). There are numerous applications in which only partial information about outcomes is available [8]. Correspondingly, the superset learning problem has received increasing attention and has been studied by various authors in recent years, albeit under different names [4, 7, 9, 2].

Setting

Consider a standard setting of supervised learning with an input (instance) space \mathcal{X} and an output space \mathcal{Y} . The goal is to learn a mapping from \mathcal{X} to \mathcal{Y} that captures, in one way or the other, the dependence of outputs (responses) on inputs (predictors). The learning problem essentially consists of choosing an optimal model (hypothesis) M^* from a given model space (hypothesis space) \mathbf{M} , based on a set of training data

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

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

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

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

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

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

Generalized Loss Minimization

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

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

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

In [6], we developed a generic approach to superset learning that can be seen as a generalization of empirical risk minimization. More specifically, this approach is based on the minimization of the empirical risk with respect to the generalized loss function or *optimistic superset loss* (OSL)

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

instead of the original loss L . Thus, each candidate model $M \in \mathbf{M}$ is evaluated in terms of

$$\bar{\mathcal{R}}_{emp}(M) = \frac{1}{N} \sum_{n=1}^N L^*(Y_n, M(\mathbf{x}_n)) , \quad (6)$$

and an optimal model M^* is one that minimizes (6). The choice of the minimum as an aggregation of the possible true losses in (5) is motivated by the goal of “data disambiguation”, i.e., of finding the most plausible instantiations $y_n^* \in Y_n$ of the ambiguous observations Y_n . For details of this approach, we refer to [?].

Interestingly, several existing machine learning methods are recovered as special cases of our framework, i.e., for specific combinations of output space, loss function and imprecisiation of the data. For example, support vector regression [10] is obtained as a generalisation of standard regression with L_1 loss if precise output values $y_n \in \mathbb{R}$ are replaced by interval-valued data $Y_n = [y_n - \epsilon, y_n + \epsilon]$, i.e., ϵ -intervals around the original data points; in fact, our generalized loss then corresponds to the ϵ -insensitive loss function used in support vector regression.

Locally Weighted Linear Regression

Obviously, the OSL L^* is a relaxation of the original loss L in the sense that $L^* \leq L$. More specifically, the larger the set Y , the smaller the loss:

$$Y \supset Y' \Rightarrow \forall \hat{y} \in \mathcal{Y} : L^*(Y, \hat{y}) \leq L^*(Y', \hat{y})$$

Thus, the loss $L(y, \hat{y})$ incurred for a prediction \hat{y} can be weakened by replacing the original observation y with a subset around y , and the larger the subset, the smaller the loss. This observation is on the basis of our idea of realizing locally weighted inference within the framework of superset learning.

In particular, our framework suggests natural approaches to locally weighted linear and support vector regression that deviate from the standard approaches [1, 3]. In standard locally weighted regression, a prediction \hat{y} for a query instance \mathbf{x} takes the form $\hat{y} = \mathbf{x}^\top \boldsymbol{\beta}^*$, where $\boldsymbol{\beta}^*$ is obtained by minimizing a sum of weighted losses:

$$\boldsymbol{\beta}^* = \arg \min_{\boldsymbol{\beta}} \sum_{n=1}^N k(\mathbf{x}, \mathbf{x}_n) L(y_n, \mathbf{x}_n^\top \boldsymbol{\beta}) \quad , \quad (7)$$

where $k(\mathbf{x}, \cdot)$ is a kernel function that assigns large weights to instances \mathbf{x}_n close to \mathbf{x} and smaller weights to instances farther away; moreover, L is a loss function such as absolute or squared difference.

Instead of *weighing* each individual loss $L(y_n, \hat{y}_n)$ in terms of a constant factor $c_n = k(\mathbf{x}, \mathbf{x}_n)$, our approach suggests another modification, namely, a specific kind of “stretching” of the loss function around the observed outcome y_n , which is achieved by the OSL (5) if y_n is replaced by a superset $Y_n \ni y_n$. This superset reasonably takes the form of an interval $[y_n - \delta_n, y_n + \delta_n]$, where the length δ_n plays the role of the weight $c_n = k(\mathbf{x}, \mathbf{x}_n)$ in the original approach. Thus, our method finds the generalized empirical risk minimizer

$$\boldsymbol{\beta}^* = \arg \min_{\boldsymbol{\beta}} \sum_{n=1}^N L^*(Y_n, \mathbf{x}_n^\top \boldsymbol{\beta}) \quad . \quad (8)$$

This is accomplished by an iterative algorithm that alternates between two steps:

- Given a current *instantiation* of the set-valued data, i.e., values $y_n^* \in Y_n$, $n = 1, \dots, N$, a parameter vector $\boldsymbol{\beta}^*$ is fit to this data using standard linear regression (the first instantiation is initialized with the original data y_n , i.e., the midpoints of the intervals Y_n).

- Then, given this solution, an improved instantiation is determined by replacing the current values y_n^* with those that appear most plausible under this solution:

$$y_n^* \leftarrow \arg \min_{y \in Y_n} L(y, \mathbf{x}_n^\top \boldsymbol{\beta}^*)$$

For this algorithm, convergence to the optimal solution (8) can be proved formally. The values y_n^* eventually found serve as a (hypothetical) *disambiguation* of the set-valued data Y_n , $n = 1, \dots, N$.

Experiments

In a first experimental study, we compared our interval-based approach to locally weighted linear regression with the conventional one on a number of UCI benchmark data sets. The loss function L in (7) was initialised with the standard squared error loss, and the Gaussian kernel function $k(\mathbf{x}, \mathbf{x}') = \exp(-\lambda^2 \|\mathbf{x} - \mathbf{x}'\|^2)$ was used. To assure maximal comparability of the two approaches, we used a similar function, namely $\exp(\lambda^2 \|\mathbf{x} - \mathbf{x}'\|^2) - 1$, to specify the width of intervals in our method. Prediction accuracy (mean squared error) was estimated by means of a 10-fold cross validation, and the hyper-parameter λ was selected in an internal (5-fold) cross validation.

Table 1 shows the results in terms of average error \pm standard deviation. As can be seen, the two methods perform more or less on a par; at least, there are no statistically significant differences between them.

Table 1: Experiment result on real-world data (with standardized outputs).

data set	locally weighted	interval-based
Breast cancer Wisconsin	0.1399 \pm 0.4781	0.1432 \pm 0.4621
Red wine quality	0.6083 \pm 0.9737	0.5936 \pm 0.9601
White wine quality	0.6456 \pm 1.3481	0.6291 \pm 1.1302
Community violence pred.	0.0361 \pm 0.0735	0.0419 \pm 0.0704
Combined cycle power plant	0.0612 \pm 0.0944	0.0648 \pm 0.0933
Parkinsons telemonitoring	0.3765 \pm 0.9295	0.3596 \pm 0.6646
Physicochemical properties of protein tertiary structure	0.1011 \pm 0.3913	0.1582 \pm 0.7911

Concluding Remarks

Overall, the results are quite promising, suggesting that our method based on superset learning provides a viable alternative to standard locally weighted learning. This provides a strong motivation for investigating this idea in more detail, not only from an empirical but also from a theoretical and algorithmic point of view.

Currently, we are elaborating on locally weighted linear regression with L_1 instead of L_2 loss, a case for which our framework suggests an alternative approach to locally weighted support vector regression. Of course, going beyond local regression, the same framework can also be applied to generalize any other type of instance weighing in machine learning. Thus, building on the results so far, there are various interesting lines of research to be explored in future work.

References

- [1] W.S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368), 1979.
- [2] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12:1501–1536, 2011.
- [3] E.E. Elattar, J.Y. Goulermas, and Q.H. Wu. Electric load forecasting based on locally weighted support vector regression. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 40(4):438–447, 2010.
- [4] Y. Grandvalet. Logistic regression for partial labels. In *IPMU-02, Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 1935–1941, Annecy, France, 2002.
- [5] E. Hüllermeier. Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization. *International Journal of Approximate Reasoning*, 55(7):1519–1534, 2014.
- [6] E. Hüllermeier and W. Cheng. Superset learning based on generalized loss minimization. In *Proceedings ECML/PKDD-2015, European Conference on Machine Learning and Knowledge Discovery in Databases*, Porto, Portugal, 2015.

- [7] R. Jin and Z. Ghahramani. Learning with multiple labels. In *16th Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, 2002.
- [8] L.P. Liu and T.G. Dietterich. A conditional multinomial mixture model for superset label learning. In *Proc. NIPS*, 2012.
- [9] N. Nguyen and R. Caruana. Classification with partial labels. In *Proc. KDD 2008, 14th Int. Conf. on Knowledge Discovery and Data Mining*, Las Vegas, USA, 2008.
- [10] B. Schölkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

Fuzzybasierte Auslegung von Wegenetzen für fahrerlose Transportsysteme

M. Sc. Sarah Uttendorf, Prof. Dr.-Ing. Ludger Overmeyer

IPH - Institut für Integrierte Produktion Hannover gGmbH
Hollerithallee 6, 30419 Hannover
E-Mail: uttendorf@iph-hannover.de

Einführung

Durch die stetige Weiterentwicklung von fahrerlosen Transportsystemen (FTS) werden sie in immer mehr Anwendungsbereichen eingesetzt. Hierdurch nimmt auch das Aufgabenspektrum der fahrerlosen Transportfahrzeuge (FTF) zu und sie müssen immer komplexere Aufgaben erfüllen. Dieses wirkt sich auch unmittelbar auf die Planung von FTS aus, indem der Auslegung des Wegenetzes, insbesondere bei komplexen Aufgabenstellungen, eine verstärkte Bedeutung zukommt.

Die Erstellung von Wegenetzen für FTS ist ein extrem zeitaufwändiger Prozess, der durch überwiegend manuelle Schritte geprägt ist. Die Systemplaner müssen mehrere verschiedene Inputfaktoren, wie bspw. die Transportmatrix, das Hallenlayout und die FTF-Eigenschaften, gleichzeitig berücksichtigen und bewerten, um ein optimales Wegenetz zu generieren. Die Systemplaner verfügen über großes Erfahrungswissen darüber, welche Wegenetze anwendbar und effizient sind.

Um diesen Planungsaufwand zu reduzieren und den Systemplanern eine Unterstützung zu bieten, wäre ein System notwendig, das automatisiert effiziente und in der Praxis anwendbare Wegenetze generiert. Eine manuelle Auslegung der Wegenetze ist allerdings sehr zeitintensiv und bietet sich nur für kleine Anwendungsumgebungen an. Ziel ist es daher ein System zu erschaffen, das automatisiert und schnell effiziente und anwendbare Wegenetze generiert.

In der Mathematik gibt es einige Ansätze, um automatisiert Wegenetze für unterschiedliche Probleme zu erstellen. Allerdings liefern diese Wegenetzalgorithmen Lösungen, die zwar mathematisch sinnvoll, aber auf ein reales Hallenlayout nicht ohne manuelle Anpassungen anwendbar sind. Die manuelle Planung hingegen liefert Lösungen, die auch praktisch auf ein Layout anwendbar sind. Dabei werden oft längere Wege zugunsten einer geringeren Kollisionsgefahr präferiert oder Wege zusammengeführt um Platz zu sparen. Da jedoch viele Optimierungsprobleme

gleichzeitig zu beachten sind, kann ein Systemplaner nur schwer ein optimales Wegenetz für eine größere Einsatzumgebung erstellen.

Dieser Beitrag stellt ein Expertensystem vor, das automatisiert Wegenetze erstellen kann, die sowohl effizient als auch in der Praxis anwendbar sind. Das Expertenwissen wurde dazu in Form einer Fuzzy-Logik gespeichert und mit etablierten Wegenetzalgorithmen kombiniert.

Dieser Beitrag gliedert sich wie folgt: Zunächst werden grundlegende Konzepte vorgestellt, die zum Verständnis dieses Beitrags relevant sind. Anschließend wird die Gesamtstruktur des Systems erläutert. Das nachfolgende Kapitel beschreibt kurz die Funktionsweise der Wegenetzalgorithmen und nachfolgend wird der erstellte Fuzzy-Logik-Controller (FLC) und das Zusammenspiel mit den Wegenetzalgorithmen erklärt. Das nächste Kapitel zeigt die Ergebnisse des Expertensystems und beinhaltet eine Diskussion dieses. Das letzte Kapitel enthält die Zusammenfassung und den Ausblick auf weitere Schritte.

Grundlegende Konzepte

Wegenetze für fahrerlose Transportsysteme

FTS sind Materialtransportsysteme, die im automatisierten Fertigungsumfeld den Transport zwischen verschiedenen Stationen übernehmen [1]. Die Aufgaben, die die Fahrzeuge übernehmen können werden immer komplexer und dadurch vergrößert sich auch das Einsatzfeld auf denen sie anwendbar sind. Dies erschwert die manuelle Planung der Wegenetze, da die Systemplaner immer mehr Inputs berücksichtigen müssen.

Zur Erstellung von Wegenetzen für FTS existieren verschiedene Ansätze. [2] unterscheidet zwischen einem „fixed-path“-Ansatz, in dem die Wegenetze physikalisch im Hallenboden definiert sind. Dazu können z. B. Magnetstreifen verwendet werden. Obwohl dieser Ansatz nicht flexibel ist und keine Alternativwege im Falle von Deadlocks oder Kollisionen generiert werden können, liegt der Hauptvorteil in der niedrigen Systemkomplexität. Er bietet sich an, wenn die Größe des Hallenlayouts überschaubar ist und keine Änderungen in den Positionen der Stationen zu erwarten sind.

Ein weiterer Ansatz ist der „open-path“-Ansatz, bei dem keine physikalische Repräsentation der Wegenetze benötigt wird. Die Informationen über die Wege

sind entweder in den Fahrzeugen selbst oder in einer zentralen Intelligenz gespeichert. Dadurch ist der Standort der Fahrzeuge zu jedem Zeitpunkt bekannt und Alternativwege können schnell übermittelt werden.

Im dritten Ansatz, der autonomen Navigation, entscheiden die Fahrzeuge selbst über ihren Weg. Viele Einflussfaktoren, wie z. B. Personenverkehr oder andere Fahrzeuge können dabei allerdings die optimale Wegefindung stören.

Das in diesem Beitrag vorgestellte System konzentriert sich auf die ersten beiden Ansätze, den „fixed-path“ und den „open-path“-Ansatz.

Algorithmen zur Auslegung von Wegenetzen für Fahrzeuge

Im Bereich der automatisierten Erstellung von Wegenetzen sind vor allem die etablierten Wegenetzalgorithmen von Bedeutung. Dabei ist grundlegend zwischen Wegenetzalgorithmen zu unterscheiden, die den Fokus haben, die Berechnungszeit zu verkürzen (z. B. [3, 8]) oder den kürzesten Weg zu ermitteln (z. B. [6, 7]).

[8] verwendet einen A*-Algorithmus mit einigen Modifikationen (Phi*, Basic Theta*), um die Berechnungszeit für die Generierung von Wegenetzen zu minimieren.

[6] minimiert hingegen die Transportzeit zwischen zwei Stationen im Layout. Basierend auf einem dynamischen Algorithmus entwickelt er eine optimale Lösung, indem jeder Zelle im Layout ein gewisser Risikofaktor zugewiesen wird, der die Wahrscheinlichkeit für bspw. eine Kollision in diesem Bereich determiniert.

Alle Ansätze erstellen zwar hinsichtlich ihres jeweiligen Optimierungsfokus effiziente Wegenetze, allerdings sind diese nicht auf ein reales Hallenlayout anwendbar. Gründe hierfür sind z. B. die ineffiziente Ausnutzung von Freiräumen sowie eine fehlende Risikobetrachtung von möglichen Deadlocks und Kollisionen.

Ein Ansatz, die manuelle Auslegung von Wegenetzen teilweise zu automatisieren, ist die Software iGoEasy [5] der Still GmbH. Hiermit ist es dem Benutzer möglich, die Wegenetze an einem Tablet zu planen. Der Anwender fährt mit dem Fahrzeug den Weg entlang, den das spätere FTF fahren soll. Dieser Weg wird aufgezeichnet und dem Benutzer zur Nachbearbeitung auf dem Tablet zur Verfügung gestellt. Nachteil dieses Ansatzes ist, dass der Benutzer die Wege im Vorfeld schon geplant haben muss und daher ist der Ansatz nur auf kleinen Produktionsumgebungen anwendbar.

Ablauf der fuzzy-unterstützten Wegenetzgenerierung

Um die im ersten Kapitel genannten Herausforderungen zu bewältigen und eine automatisierte Generierung von anwendbaren und effizienten Wegenetzen zu ermöglichen, wurde ein Hybrid aus etablierten Wegenetzalgorithmen und menschlichem Erfahrungswissen in Form einer Fuzzy-Logik entwickelt. Abbildung 1 zeigt eine Übersicht des Ablaufs der fuzzy-unterstützten Wegenetzgenerierung.

Anschließend bewerten zwei modifizierte Wegenetzalgorithmen die einzelnen Zellen basierend auf ihren Heuristiken. Zellen die günstig sind werden geringe Kosten zugewiesen und Zellen die ungünstig sind, weil sie bspw. Hindernisse oder Stationen enthalten, werden extrem hohe Kosten zugewiesen. Von den Algorithmen werden Kostenfunktionen erstellt, die anschließend den günstigsten Weg von der Startstation s_1 zur Zielstation s_2 definieren.

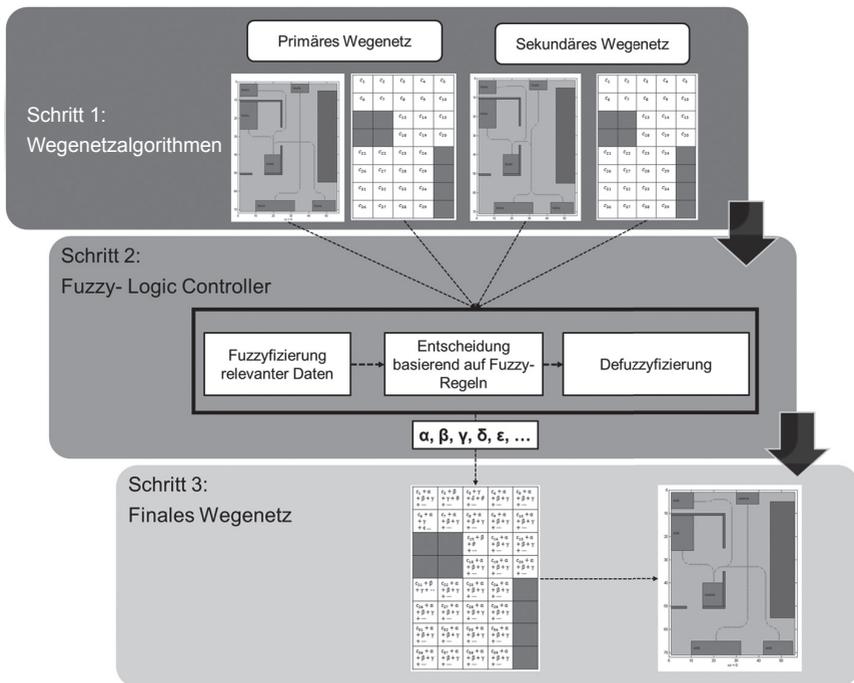


Bild 1: Vorgehensweise der fuzzy-unterstützten Wegenetzgenerierung

In *Schritt 1* wird das Hallenlayout der Einsatzumgebung der FTF eingelesen, vorverarbeitet und in ein Gitter mit äquivalent großen Zellen unterteilt.

Basierend auf den Heuristiken der beiden modifizierten Wegenetzalgorithmen werden unterschiedliche Wegenetze mit unterschiedlichen zu Grunde liegenden Kostenmatrizen generiert.

In *Schritt 2* evaluiert ein Fuzzy-Logik-Controller (FLC) die übergebenen Wegenetze und Kostenmatrizen. Der FLC wurde basierend auf Expertenbefragungen erstellt und wird in Kapitel 5.1 näher erläutert. Hierbei werden relevante Inputdaten (z. B. Anzahl der Kreuzungen, Weglänge, Anzahl der Kurven) sowie die Kostenmatrix evaluiert und fuzzyfiziert. Dabei werden den Inputdaten linguistische Terme basierend auf Zugehörigkeitsfunktionen zu gewissen Klassen zugewiesen. Auf Basis der durch Expertenwissen erstellten Regeln werden die Wegenetze auf ihre Anwendbarkeit und Effizienz hin evaluiert. Verschiedene Regeln können zur selben Zeit feuern. Die Regeln liefern Output in Form von Anweisungen, ob die Kosten in bestimmten Wegebereichen erhöht werden sollten und zu welchem Grad. Anschließend erfolgt die Defuzzifizierung des Regeloutputs, indem der Handlungsanweisung ein numerischer Wert $\alpha, \beta, \gamma, \dots$; $\alpha, \beta, \gamma \in \mathbb{R}$ zugeordnet wird.

In *Schritt 3* werden die generierten Werte $\alpha, \beta, \gamma, \dots$ an spezifizierten Stellen auf die Kostenmatrizen addiert. Eine erneute Berechnung des günstigsten Weges von der Startstation s_1 zur Zielstation s_2 liefert das fuzzy-optimierte Wegenetz.

Eingesetzte Wegenetzalgorithmen

Als Grundlage für die fuzzy-unterstützte WegenetzAuslegung dienen Wegenetzalgorithmen. Das wichtigste Kriterium bei der Auswahl der Algorithmen, war die Voraussetzung, dass die Wegenetzalgorithmen das Produktionslayout in ein fest definiertes Raster mit gleich großen Zellen unterteilen und den Zellen Kosten zuweisen. Für das entwickelte Expertensystem wurden der A* und der Lee Algorithmus ausgewählt.

Sowohl der A*- als auch der Lee-Algorithmus unterteilen die zu untersuchende Fläche in ein fest definiertes Gitter. Im vorliegenden Falle sind die Zellen quadratisch. Jeder Zelle wird ein fester Kostenwert zugewiesen. Objekten (z. B. Wände, Stationen) werden hierbei sehr hohe Kosten zugewiesen, so dass es nie zu einer Wegegenerierung in diesem Bereich kommt. Günstigen Wegabschnitten werden niedrige Kosten zugewiesen. Die Algorithmen stellen Kostenfunktionen

auf und minimieren, die Kosten von der Startstation s_1 zur Zielstation s_2 . Die Kostenfunktion besteht aus zwei Teilen:

$$f(x) = g(x) + h(x) \tag{1}$$

mit $h(x)$ als Heuristik um die Kosten von der aktuellen Zelle x zu s_2 zu schätzen. Die geschätzten Kosten dürfen hierbei nie die tatsächlichen Kosten überschreiten. Als Heuristik wurde z. B. die euklidische Distanz gewählt:

$$h(x) = \sqrt{(x_x - x_{s2})^2 + (y_x - y_{s2})^2} \tag{2}$$

$g(x)$ beschreibt die bis dahin existierenden Kosten von s_1 zu x .

$$g(x) = \begin{cases} x.pre.costs + \sqrt{2c_i}(x) & \text{for diagonal cells} \\ x.pre.costs + c_i(x) & \text{else} \end{cases} \tag{3}$$

In dieser Formel entspricht $c_i(x)$ dem spezifischen Kostenwert für die aktuelle Zelle x . $c_i(x)$ ist abhängig von der Eignung der Zelle für einen optimalen Weg und wird durch die Kostengewichtung durch den Wegenetzalgorithmus bestimmt. Der eigentliche Kostenwert ist ein fester numerischer Wert $c_i(x) = \text{konstant}$. Eine genauere Beschreibung der ausgewählten Wegenetzalgorithmen ist in [11] zu finden.

Speicherung des Expertenwissens in Fuzzy-Logik

Fuzzy-Logik wurde erstmals von Lotfi Zadeh 1965 an der University of Berkley eingeführt [12]. Fuzzy-Logik wurde als Methode für die Abbildung des Expertenwissens in der Wissensbasis des vorgestellten Expertensystems ausgewählt, da sie in der Lage ist, die Unschärfe von Variablen zu erfassen und zu speichern. Eine Variable wird nicht wie im klassischen binärischen Ansatz als $x \in [0, 1]$ ausgedrückt, sondern x kann graduell einem Attribut angehören, also $x \in (0, 1)$. Die Fuzzy-Logik bietet für dieses Expertensystem den Hauptvorteil, mit linguistischen Variablen zu arbeiten. Menschliche Experten sind in der Lage, zielgeführte Entscheidungen aufgrund ihres Wissens zu treffen. Allerdings ist die Art der Wissensrepräsentation unscharf. Eine Aussage, wie „Wenn viele Kreuzungen da sind und das Wegenetz an dieser Stelle sehr stark ausgelastet ist, dann lege einen Alternativweg an.“ beinhaltet Entscheidungskompetenz ist aber schwer in der traditionellen Informatik umzusetzen. Die Fuzzy-Logik ist in der Lage, mit

Begriffen wie „viele Kreuzungen“ oder „stark ausgelastet“ zu arbeiten und die Unschärfe zu speichern und verwertbar zu machen. Eine vertiefte Erklärung zur Nutzung der Fuzzy-Logik im Expertensystem ist in [10] zu finden.

Aufbau eines Fuzzy-Logik-Controllers basierend auf Expertenwissen

Um das Wissen der Systemexperten in einer fuzzy-basierten Form zu speichern, ist die Extrahierung des Wissens in fuzzy-basierter Form notwendig. Eine Herausforderung bei der Erstellung dieser Art von Befragungen ist, dass keine herkömmliche Likert-Skala verwendet werden kann. Bei der Likert-Skala werden dem Befragten zwei entgegengesetzte Aussagen zur Auswahl gestellt und der Befragte hat die Möglichkeit in vordefinierten Abstufungen seine Zustimmung zu einer Aussage zu geben. Meist wird eine Äquidistanz zwischen den Stufen gewählt, d.h. gleiche Abstände zwischen den Stufen.

Diese Befragungsmethode ist aber nicht in der Lage, das Erfahrungswissen in seiner Unschärfe zu speichern. Dazu wurde am European Centre for Soft Computing das fuzzy-basierte Frage-Tool Quale [9] entwickelt. Die Software kann dazu genutzt werden, die Unschärfe des Wissens zu extrahieren und zu repräsentieren [4]. Um das Wissen der Systemplaner für FTF zu extrahieren, wurde speziell ein fuzzy-basiertes Interview entwickelt. Dabei werden den Experten 30 verschiedene Layouts gezeigt und sie müssen verschiedene Inputs bewerten. Tabelle 1 zeigt einen Auszug aus der Liste der Inputs. Bei dieser Art von Befragung muss kein exakter Wert angegeben werden, sondern der Befragte gibt auf einer ersten linguistischen Skala an, welchen Wert eine Variable auf dem jeweiligen Layout annimmt. Mittels einer zweiten Skala unterhalb gibt er zusätzlich an, bis zu welchem Bereich er der Aussage nach oben und unten hin noch zustimmt.

Tabelle 1: Inputfaktoren zur Expertenbefragung

Input	Linguistische Reichweite	Information
number of path crossings	min= too few, max= too many	Are there (too few/too many) path crossings?
number of change of directions	min= too few, max= too many	Are there (too few/ too many) changes of directions?
amount of free space	min= too few, max= too many	Is there (too little/ too much) free space?
amount of merged paths	min= too little, max= too many	Are there (too little/ too many) merged paths?
length of all the paths combined	min= too short, max= too long	Is the overall length of the paths in the roadmap (too short/ too long)?
distance between paths	min= too narrow, max= too big	Is the distance between paths (too narrow / too big)?
risk of collision at turning/merging/-crossing positions	min= very low, max= very high	Is the risk of collision between different AGVs (very low / very high)?
number of paths	min= too many, max= too few	Are there (too many / too few) paths in the layout?
distance to walls	min= sufficient, max= not sufficient	Is the distance to the walls (sufficient / not sufficient)?

Aus den Antworten der Befragung werden die Zugehörigkeitsfunktionen der Inputs zu den Attributen generiert. Auf diese Weise entsteht als weiteres Ergebnis der Befragung eine Regelbasis. Tabelle 2 zeigt Beispiele für diese Regeln.

Tabelle 2: Auszug aus der Regelbasis des Fuzzy-Logik-Controllers

Re- gel	Optimierungs- problem	Input	Fuzzy Regel	Out- put
1	Minimiere Anzahl Kreuzungen	$\#crossings$	IF $\#Crossings_{modA}$ high AND $\#Crossings_{modLee}$ low THEN $cost_{overall_{modA}}$ NOT low AND $cost_{overall_{modLee}}$ low	$\alpha \in$ [0, 100]
2	Minimiere Kollisionsri- siko	$\#transports,$ $\#merged-$ $cells$	IF $\#transports$ very high AND $\#mergedcells$ high THEN costs high.	$\beta \in$ [0, 100]
...
n	Minimieren der Weglän- ge und der Anzahl der Wendepun- kte	Path len- gths, $\#$ turning points	IF pathlength short AND $\#$ turning points few THEN costs low	$\gamma \in$ [0, 100]

Interaktion des Fuzzy-Logik-Controllers mit den etablierten Wegenetzalgorithmen

Um die Outputs der Regeln in die Auslegung der Wegenetze einfließen lassen zu können, werden die Outputs nach der Defuzzifizierung mit den Ergebnissen der Wegenetzalgorithmen kombiniert. Wie im vorherigen Kapitel beschrieben, unterteilen die Wegenetzalgorithmen das Hallenlayout in ein fest definiertes Gitter.

Jedem Gitterpunkt wurde ein Kostenwert c_i zugewiesen. Nachdem der FLC die defuzzifizierten Outputs in der Form $\alpha, \beta, \gamma, \dots$ an den Wegenetzalgorithmus übermittelt hat wird die Kostenmatrix neu berechnet. Wie in Abbildung 2 zu sehen ist, wird jeder Zelle ein neuer Kostenwert c_{i*} basierend auf der Einschätzung des FLC zugewiesen. c_{i*} ist damit nicht mehr konstant sondern eine Gleichung aus:

$$c_{i*} = c_i + \alpha + \beta + \gamma + \dots \quad (4)$$

Anschließend wird durch die Minimierung der Kostenfunktion $f(x)$ erneut der kostengünstigste Weg von einer Startstation s_1 zu einer Zielstation s_2 ermittelt.

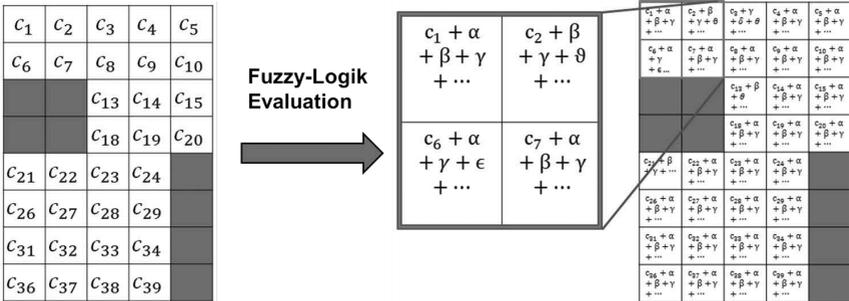


Bild 2: Fuzzy-Bewertung der Kostenmatrix

Anwendung und Ergebnisbewertung

Das vorgestellte Expertensystem ermöglicht die Verbindung von mathematischer Optimierung und menschlicher Entscheidungskompetenz. Für verschiedene Layouts wurden mit dem Expertensystem Wegenetze ausgelegt. Die nachfolgenden Abbildungen vergleichen jeweils die durch herkömmliche Wegenetzalgorithmen (hier: A*) erstellten Wegenetze mit den durch das fuzzy-unterstützte Expertensystem erstellten Wegenetzen.

In Bild 3. ist links das vom A*-Algorithmus und rechts das vom Expertensystem erstellte Wegenetz zu sehen. Hieraus ist ersichtlich, dass die Anzahl der diagonalen Wege reduziert wurde. Hierdurch hat sich zwar die Gesamtweglänge erhöht,

allerdings wurde das Risiko für Verkehrsbehinderungen, falls bspw. ein Fahrzeug oder eine Person den Weg kreuzen muss, verringert. Weiter ist bspw. der Weg von 1 nach 6 im rechten Layout deutlich länger als im linken. Das Expertensystem hat sich in diesem Fall zu Gunsten eines geringeren Risikos von Kollisionen und einem möglichen Alternativweg, falls Fahrzeuge auf dem anderen Streckenabschnitt stehen bleiben, gegen den kürzeren Weg entschieden. Die Berechnungszeit für das vom Expertensystem erstellte Wegenetz betrug 141,61s.



Bild 3: Vergleich zwischen zwei Wegenetzen, erstellt durch den A*-Algorithmus (links) und das Expertensystem (rechts)

Ein weiteres Beispiel ist in Bild 4 zu sehen. Hier ist deutlich zu erkennen, dass der Fokus des A*-Algorithmus auf der Minimierung der Weglänge liegt, während das Expertensystem auf die praktische Anwendbarkeit achtet. Links ist die Freifläche durch diagonal angeordnete Wege deutlich reduziert, während das Expertensystem Wege zusammenfügt, um die anderweitig nutzbare Freifläche zu maximieren. Die Berechnungszeit für dieses Layout betrug 9,58 s

In Bild 5 ist zusätzlich zu sehen, dass auch zu Gunsten geteilter Wege (Weg 1-3) geringfügig längere Wege gewählt werden, um den Freiraum zu maximieren. Das Risiko für Kollisionen zwischen Fahrzeugen wurde zudem auch minimiert, da die Anzahl von Wegkreuzungen reduziert wurde. Auch dies hat zur Folge,

dass bestimmte Wegstrecken vom Expertensystem zwar länger ausgelegt werden, allerdings bewerteten die Experten die Gefahr von Kollisionen höher als der Nutzen kürzerer Wege. Die Generierungszeit des Wegenetzes betrug 137,91s.

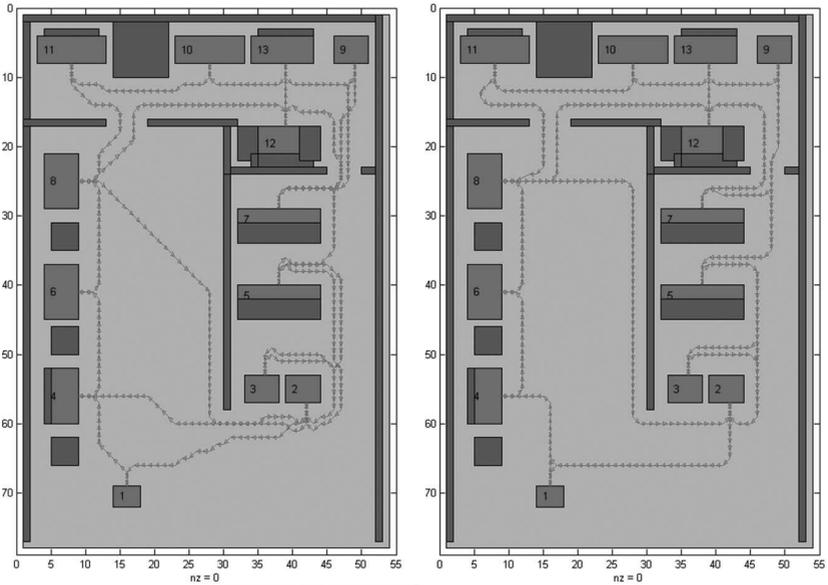


Bild 4: Vergleich zwischen zwei Wegenetzen, erstellt durch den A*-Algorithmus (links) und das Expertensystem (rechts) (II)

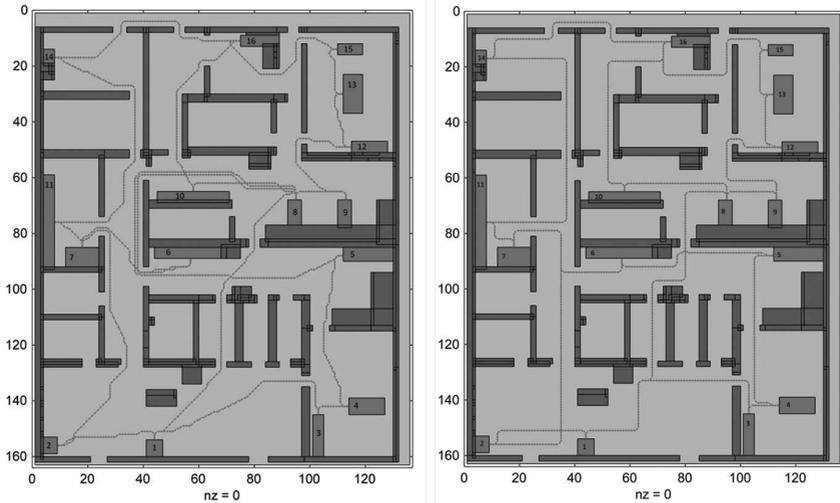


Bild 5: Vergleich zwischen zwei Wegenetzen, erstellt durch den A*-Algorithmus (links) und das Expertensystem (rechts) (III)

Trotz der gezeigten verbesserten Anwendbarkeit der Wegenetze fehlt für einen praktischen Einsatz noch die Einbeziehung der fahrzeugspezifischen Anfahrtswege. Abhängig vom Fahrzeug müssen Anfahrtskurven und Wenderadien zum Rangieren bei der Planung mit einbezogen werden. Dazu müssen verschiedene Fahrzeugtypen mit den zugehörigen Kinematikdaten in einer Datenbank hinterlegt und mit in die Planung einbezogen werden. Die Einbeziehung der Kinematikdaten wird sich stark auf den Verlauf der Wegenetze auswirken, da vor allem im Hinblick auf Kollisionsgefahren bei der Zusammenlegung von Wegen an Anfahrtspunkten eine andere Bewertung stattfinden wird.

Zusammenfassung und Ausblick

In diesem Beitrag wurde eine Lösung zur automatischen Erstellung von Wegenetzen für FTS vorgestellt. Die Wegenetze sollten dabei schnell erstellt werden und die Ergebnisse in der Praxis anwendbar sein. Obwohl die gezeigten Layouts deutlich anwendungsbezogener sind als die von bisherigen Wegenetzalgorithmen erstellten, muss noch in Simulationen gezeigt werden, dass die Wegenetze genauso

belastbar sind wie manuell erstellte. Vor allem das Verhalten der Wegenetze auf Kollisionen und ein erhöhtes Transportaufkommen wird für die praktische Einsatzbarkeit des Expertensystems von Bedeutung sein. Dazu werden im nächsten Schritt Simulationen mit der Software Sim3D und PlantSimulation durchgeführt, die eine Einsatzumgebung modellieren und die Materialflüsse simulieren.

Danksagung

Das IGF-Vorhaben (18007 N/1) der Bundesvereinigung Logistik (BVL) e.V. wird über die AiF im Rahmen des Programms zur Förderung der Industriellen Gemeinschaftsforschung (IGF) vom Bundesministerium für Wirtschaft und Energie aufgrund eines Beschlusses des Deutschen Bundestages gefördert.

Literatur

- [1] W. Maxwell, J. Muckstadt: „Design of automated guided vehicle systems“. In: IIE Trans 14, S.114-124. 1982.
- [2] H. Martínez-Barberá, D. Herrero-Pérez: „Programming multirobot applications using the ThinkingCap-II framework“. In: Advanced Engineering Informatics 24 (2010), Nr. 1, S. 62-75. ISSN 1474-0346. 2010.
- [3] R. Hall: „The fastest path through a network with random time-dependent travel times“. In: Transportation Science, Vol.20, No.3. 1986.
- [4] J. Alonso, D. Pancho, L. Magadalena, D. Nunez, D. Sánchez, P. Suárez, J. Mingot, V. Iglesias: „Quale ®: A new Toolbox for Quantitative and Qualitative Analysis of Human Perceptions“. In: 16th World Congress of the International Fuzzy Systems Association, 978-94-62520-77-6, Atlantis Press. 2015.
- [5] <http://www.still.de/igoeasy.0.0.html> . Zugriff: 15. August 2015.
- [6] K. Bae, Y. Kim; J. Han: „Finding a risk-constrained shortest path for an unmanned combat vehicle.“. In: Computers & Industrial Engineering, Volume 80, S.245-253. 2015.
- [7] C. Kim, J. Tanchoco: „Conflict-free shortest-time bidirectional AGV routing“. In: Int. J. Prod., Vol. 29, No.12, 2377-2391. 1991.

- [8] F. Duchon, A. Babinec, M. Kajan, P. Beno, M. Florek, T. Fico, L. Jurisica: „Path planning with modified a star algorithm for a mobile robot“. In: Procedia Engineering, Volume 96, S. 59-69. 2014.
- [9] <http://www.softcomputing.es/quale/> . Zugriff: 26. August 2015.
- [10] S. Uttendorf, G. Ullmann, L. Overmeyer „Automatisierte Auslegung von Wegenetzen für Fahrerlose Transportsysteme – Ein fuzzybasierter Ansatz zur optimalen Wegenetzplanung“. In: Industrie Management, GITO Verlag, 1/2015, H. 31, S. 48-52. 2015.
- [11] S. Uttendorf, L. Overmeyer: „Fuzzy-enhanced path-finding algorithm for AGV roadmaps“. In: 16th World Congress of the International Fuzzy System Association (IFSA), 9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT), 2015 Atlantis Press. 2015.
- [12] L. Zadeh: „Fuzzy Sets“. In: Information and Control, Volume 8, S. 338-353

Ein Beitrag zur robusten Beobachtersynthese für Takagi-Sugeno Modelle mit nichtmessbaren Prämissenvariablen

Horst Schulte

Fachbereich Ingenieurwissenschaften – Energie und Information,
Fachgebiet Regelungstechnik, HTW Berlin
Wilhelminenhofstraße 75A, 12459 Berlin
E-Mail: schulte@htw-berlin.de

Einführung

In dieser Arbeit wird ein verallgemeinerter Beobachterentwurf zur Zustands- und Parameterschätzung für Takagi-Sugeno Modelle vorgestellt. Die Verallgemeinerung beruht darauf, dass die Messbarkeit der Prämissenvariablen nicht vorausgesetzt wird.

Mittels quadratischer Ljapunov-Funktionen wurden in [1], [2] erstmals Lineare Matrix Ungleichungen (LMI) zur Bestimmung der Beobachtermatrizen angegeben. Mit der Voraussetzung, dass die Prämissenvariablen in den Zugehörigkeitsfunktionen vollständig messbar sind, garantiert diese Art der Beobachtersynthese ein global asymptotisches Abklingen des Schätzfehlers. In vielen Anwendungen, insbesondere wenn Systemzustände und zeitvariante Parameter simultan rekonstruiert werden müssen, ist die Voraussetzung von vollständig messbaren Prämissenvariablen nicht erfüllt und damit das asymptotische Abklingen des Schätzfehlers nicht garantiert.

Um dieses Problem zu lösen wurden in den letzten Jahren verschiedene Verfahren entwickelt, welche Modellunsicherheiten hervorgerufen durch nichtmessbare Prämissenvariablen beim Stabilitätsnachweis formal berücksichtigen. Hierzu wurde in [3] eine Verfahrensweise vorgestellt, bei dem die Unsicherheiten über eine obere Normabschätzung in der Fehlerdifferentialgleichung (Fehler-Dgl.) berücksichtigt werden. Der formale Nachweis der Stabilität erfolgt mit der Existenz einer gemeinsamen quadratischen Ljapunov-Funktion mit Nebenbedingung. In [4] wird mit dem Mittelwertsatz der Differentialrechnung und der Methode der Sektor-Nichtlinearitäten die unsicherheitsbehaftete Fehler-Dgl. in ein autonomes System überführt. Der Vorteil ist, dass der Stabilitätsnachweis als Standard-LMI

Problem für Takagi-Sugeno (TS) Modelle erfolgt. Ein ähnlicher Ansatz wird in [5] und [6] für zeitdiskrete linear parameter-variable (LPV) Systeme mit unsicheren Parametern durchgeführt. In [7] wird ein relaxiertes Kriterium zum Nachweis der asymptotischen Stabilität der Fehler-Dgl. mittels quadratischer Ljapunov-Funktionen in Kombination mit Nebenbedingungen formuliert. Allerdings kann auch für die weniger konservativen Ansätze [4], [5], [6] und [7] schon bei realen Systemen mittlerer Komplexität keine Lösung für die LMI-gestützten Kriterien gefunden werden. In dieser Arbeit wird daher ein neuer, auf Sliding-Mode Techniken basierender robuster Ansatz vorgestellt.

Problemstellung

Gegeben ist ein TS Modell mit affinem Anteil und gemeinsamer Ausgangsmatrix ohne Durchgriff

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

mit $\mathbf{x} \in \mathbb{R}^n$ als Zustandsvektor, $\mathbf{u} \in \mathbb{R}^m$ als Eingangsvektor und $\mathbf{y} \in \mathbb{R}^p$ als Ausgangsvektor. Die linearen Teilsysteme $i = 1, \dots, N_r$ werden aus $\mathbf{A}_i \in \mathbb{R}^{n \times n}$, $\mathbf{B}_i \in \mathbb{R}^{n \times m}$ und $\mathbf{C} \in \mathbb{R}^{p \times n}$ gebildet. Die nichtlinearen Funktionen $h_i(\mathbf{z}) : \mathbb{R}^l \rightarrow \mathbb{R}$, $\mathbf{z} \mapsto h_i$ gewichten die linearen Teilsysteme im TS Modell zueinander und erfüllen die Konvexitätseigenschaft

$$\sum_{i=1}^{N_r} h_i(\mathbf{z}) = 1, \quad h_i(\mathbf{z}) \geq 0 \quad \forall i \in \{1, \dots, N_r\}, \quad (2)$$

mit \mathbf{z} als Prämissenvektor, welcher nichtmessbare Variablen bestehend aus Systemzuständen und externe Größen enthält. Ein Beobachter für ein System, dass mit (1) modelliert wird, wird nun wie folgt angegeben:

$$\dot{\hat{\mathbf{x}}} = \sum_{i=1}^{N_r} h_i(\hat{\mathbf{z}}) (\mathbf{A}_i \hat{\mathbf{x}} + \mathbf{B}_i \mathbf{u} + \mathbf{a}_i + \mathbf{L}_i (\mathbf{y} - \hat{\mathbf{y}})) \quad (3)$$

mit $\hat{\mathbf{z}}$ als Schätzung von \mathbf{z} . Die Differenz zwischen dem wahren und geschätzten Zustandsvektor ist der Fehler $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$. Die Fehler-Dgl. mit $\dot{\mathbf{e}} = \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}}$ folgt aus der

Substitution von $\mathbf{x}/\dot{\mathbf{x}}$ mit (1)/(3). Zur Analyse des Einflusses der nichtmessbaren Prämissenvariablen wird die Fehler-Dgl. erweitert um

$$+ \sum_{i=1}^{N_r} h_i(\hat{\mathbf{z}}) (\mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u} + \mathbf{a}_i) - \sum_{i=1}^{N_r} h_i(\hat{\mathbf{z}}) (\mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u} + \mathbf{a}_i). \quad (4)$$

Nach kurzer Umformung erhält man

$$\dot{\mathbf{e}} = \sum_{i=1}^{N_r} h_i(\hat{\mathbf{z}}) (\mathbf{A}_i - \mathbf{L}_i \mathbf{C}) \mathbf{e} + \underbrace{\sum_{i=1}^{N_r} (h_i(\mathbf{z}) - h_i(\hat{\mathbf{z}})) (\mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u} + \mathbf{a}_i)}_{\Delta = \Delta(\mathbf{z}, \hat{\mathbf{z}}, \mathbf{x}, \mathbf{u})}. \quad (5)$$

Die Analyse des Einflusses der nichtmessbaren Prämissenvariablen ergibt sich nach [3] aus der Abschätzung der maximalen Steigung der Unsicherheit

$$\mu = \max_{\mathbf{x}, \mathbf{u}, \hat{\mathbf{z}}} \left\| \frac{\partial}{\partial \mathbf{e}} (h_i(\mathbf{z}) - h_i(\hat{\mathbf{z}})) (\mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u} + \mathbf{a}_i) \right\| \quad (6)$$

mit

$$\left\| \sum_{i=1}^{N_r} (h_i(\mathbf{z}) - h_i(\hat{\mathbf{z}})) (\mathbf{A}_i \mathbf{x} + \mathbf{B}_i \mathbf{u} + \mathbf{a}_i) \right\| \leq \mu \|\mathbf{e}\|. \quad (7)$$

Vorausgesetzt wird hierbei, dass die nichtmessbaren Prämissenvariablen nur Systemzustände enthalten. Wie man leicht sieht, folgt die Konservativität dieses Verfahrens aus der fehlenden Berücksichtigung der Verläufe der Zugehörigkeitsfunktionen h_i .

Zum Nachweis der asymptotischen Stabilität von (5) wird in [3] eine quadratische Ljapunov-Funktion $V(\mathbf{e}) = \mathbf{e}^T \mathbf{P} \mathbf{e}$ mit $\dot{V}(\mathbf{e}) = \dot{\mathbf{e}}^T \mathbf{P} \mathbf{e} + \mathbf{e}^T \mathbf{P} \dot{\mathbf{e}}$ verwendet. Nach der Substitution von $\dot{\mathbf{e}}$ durch (5) folgt nach kurzer Umformung

$$\dot{V}(\mathbf{e}) = \sum_{i=1}^{N_r} h_i(\hat{\mathbf{z}}) \mathbf{e}^T \left((\mathbf{A}_i - \mathbf{L}_i \mathbf{C})^T \mathbf{P} + \mathbf{P} (\mathbf{A}_i - \mathbf{L}_i \mathbf{C}) \right) \mathbf{e} + 2\Delta^T \mathbf{P} \mathbf{e} < 0.$$

Mit Hilfe der Ungleichung $2\mathbf{x}^T \mathbf{y} \leq \mathbf{x}^T \mathbf{x} + \mathbf{y}^T \mathbf{y}$ wird der zweite Term abgeschätzt:

$$2\Delta^T \mathbf{P} \mathbf{e} \leq \Delta^T \Delta + \mathbf{e}^T \mathbf{P} \mathbf{P} \mathbf{e}$$

Unter Ausnutzung des Schurkomplements wird die Nebenbedingung des Optimierungsproblems $\mathbf{Q} - \mathbf{P} \mathbf{P} - \mu^2 \mathbf{I} > 0$ als lineare Matrixungleichung (LMI) formuliert

$$\begin{pmatrix} \mathbf{Q} - \mu^2 \mathbf{I} & \mathbf{P} \\ \mathbf{P} & \mathbf{I} \end{pmatrix} > 0, \quad \mathbf{P} > 0. \quad (8)$$

Zur gezielten Beeinflussung der Abklingrate des Beobachterfehlers wird \mathbf{Q} häufig über die Abklingrate α eingestellt mit $\mathbf{Q} = 2 \alpha \mathbf{P}$

$$\begin{pmatrix} 2\alpha \mathbf{P} - \mu^2 \mathbf{I} & \mathbf{P} \\ \mathbf{P} & \mathbf{I} \end{pmatrix} > 0, \quad \mathbf{P} > 0. \quad (9)$$

Robuste Beobachtung mit Sliding Modes

Ein Ansatz zur Reduktion der Konservativität besteht nun darin, das Fehlersystem selber als ein durch unbekannte Eingänge gestörtes System zu betrachten

$$\dot{\mathbf{e}} = \sum_{i=1}^{N_r} h_i(\hat{\mathbf{z}}) \left((\mathbf{A}_i - \mathbf{L}_i \mathbf{C}) \mathbf{e} + \mathbf{D}_i \boldsymbol{\xi} \right) \quad (10)$$

mit $\sum_{i=1}^{N_r} h_i(\hat{\mathbf{z}}) \mathbf{D}_i \boldsymbol{\xi} = \Delta(\mathbf{z}, \hat{\mathbf{z}}, \mathbf{x}, \mathbf{u})$. Unter der Annahme, dass ein Anteil der Prämissenvariablen messbar ist, lässt sich der Fehlervektor aufteilen in einen messbaren \mathbf{e}_y und nichtmessbaren Anteil \mathbf{e}_x . Unter messbare Fehler versteht man in diesem Zusammenhang Fehler, welche durch einen gewöhnlichen Takagi-Sugeno Beobachter rekonstruierbar sind und nicht durch nichtmessbare Prämissenvariablen gestört werden. Hierfür lässt sich ein Takagi-Sugeno Sliding Mode Beobachter angeben:

$$\dot{\mathbf{e}}_x = \sum_{i=1}^{N_r} h_i(\hat{\mathbf{z}}) \left(\tilde{\mathbf{A}}_i \hat{\mathbf{e}}_x - \mathbf{G}_{l,i} \mathbf{e}_y - \rho \mathbf{G}_{n,i} \frac{\mathbf{P}_2 \mathbf{e}_y}{\|\mathbf{P}_2 \mathbf{e}_y\|} \right) \quad (11)$$

Die einzelnen Schritte dieses Verfahrens werden in einer erweiterten Version dieses Beitrags im Detail dargelegt und mit zwei Beispielen aus [8] und [9] veranschaulicht.

Literatur

- [1] R. Palm und D. Driankov, Towards a systematic analysis of fuzzy observers, *18th International Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, 1999, 179–183.
- [2] K. Tanaka und H. O. Wang, Fuzzy Regulators and Fuzzy Observers: A Linear Matrix-Inequality Approach, *IEEE International Conference on Decision and Control*, 1997, 1315–1320.

- [3] P. Bergsten, R. Palm und D. Driankov. Observers for Takagi-Sugeno fuzzy systems. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 32(1):114–121, 2002.
- [4] D. Ichalal, B. Marx, D. Maquin und J. Ragot. On observer design for nonlinear Takagi-Sugeno systems with unmeasurable premise variables. *In International Symposium on Advanced Control of Industrial Processes, ADCONIP 2011*, Hangzhou, P.R. China, May 2011.
- [5] H. Hammouri und B. Uchueddu. High gain observers based on immersion technics. In 49th IEEE Conference on Decision and Control, CDC 2010, Atlanta, GA, USA, December 2010.
- [6] M. Heemels, J. Daafouz und G. Millerioux. Observer-based control of discrete-time LPV systems with uncertain parameters. *IEEE Transactions on Automatic Control*, 55(9):2130–2135, September 2010.
- [7] D. Ichalal, B. Marx, J. Ragot und D. Maquin, Advances in observer design for Takagi-Sugeno systems with unmeasurable premise variables, *20th Mediterranean Conference on Control and Automation*, 2012, 848–853.
- [8] Z. Lendek, T.M. Guerra, R. Babuska und B. De Schutter, *Stability Analysis and Nonlinear Observer Design Using Takagi-Sugeno Fuzzy Models*, Springer-Verlag Berlin Heidelberg, 2010.
- [9] A. Kroll und H. Schulte, Benchmark problems for nonlinear system identification and control using Soft Computing methods: Need and overview, *Applied Soft Computing*, 2014, Vol. 25, No. 13, 496–513.

Benchmark-Anwendung zur Schwingungsanalyse und -dämpfung von Regalbediengeräten am Beispiel eines Labormodells

Tobias Loose

Hochschule Heilbronn, Technisches Logistikmanagement
Max-Planck-Str. 39, 74081 Heilbronn
E-Mail: tobias.loose@hs-heilbronn.de

Einführung

Regalbediengeräte werden in der Logistik zur Ein- und Auslagerung von Ladeeinheiten eingesetzt. Durch die Gestaltung des Tragwerks und verhältnismäßig großen Massen ist ein Regalbediengerät schwingungsanfällig. Gerätehersteller lösen diese Problematik u.a. durch konstruktive Maßnahmen. Dennoch existieren Anlagen in der Industriep Praxis, bei denen eine aktive Maßnahme zur nachträglichen Verbesserung des Schwingungsverhaltens erforderlich ist. Insbesondere bei Paletten-Regalbediengeräten besteht der Bedarf einer Reduzierung von Schwingungen beim Anfahren von Positionen. Das Schwingen wirkt sich nachteilig auf die Materialbeanspruchung und Spielzeit aus. Die Anlagenbetreiber haben daher ein großes Interesse an nachträglichen Optimierungen.

In dieser Arbeit wird ein ebenso schwingungsfähiges Labormodell eines Regalbediengerätes vorgestellt. Das Ziel dieser Arbeit besteht darin mit leicht interpretierbaren und implementierbaren Methoden die Schwingung bei der Positionsregelung zu minimieren. Die Methoden werden zunächst am Labormodell demonstriert. In folgenden Arbeiten sollen i.S. einer Benchmark-Anwendung weitere Regler entwickelt werden. Letztlich soll eine geeignete Methode in einer realen, bestehenden Anlage zum Einsatz kommen. Bei diesen Anlagen liegt der Fokus auf einer *unkomplizierten* „plug-and-play“ Lösung, die in die bestehende Maschinensteuerung nachträglich leicht zu implementieren ist. Die Gründe dafür sind u.a. die Einhaltung der Sicherheitsstandards oder des störungsfreien Betriebs, d.h. die Anlage darf nicht wegen einer aufwändigen Inbetriebnahme mehrere Tage ausfallen. Daher wird in diesem Beitrag ein neues vereinfachtes Modell eines Regalbediengerätes vorgeschlagen. Zudem werden gängige Industrie-Praxismethoden zur Parameteridentifikation gezeigt. Derartige Methoden und Modelle sollen den Weg für eine vereinfachte Implementierung ebnen.

Zudem eignet sich das Labormodell für eine praxisorientierte Vorlesung, um sehr anwendungsnah viele regelungs- und automatisierungstechnische Methoden didaktisch darzustellen. Damit lassen sich viele typische Problemstellungen aus der Praxis zeigen und entsprechend lösen.

Aufbau und Anwendung von Regalbediengeräten

Ein Regalbediengerät (RBG) ist ein automatisch oder manuell betriebenes Fahrzeug zur Ein- und Auslagerung von Waren in bzw. aus einem Hochregallager, siehe Bild 1 und [3, 7, 10].

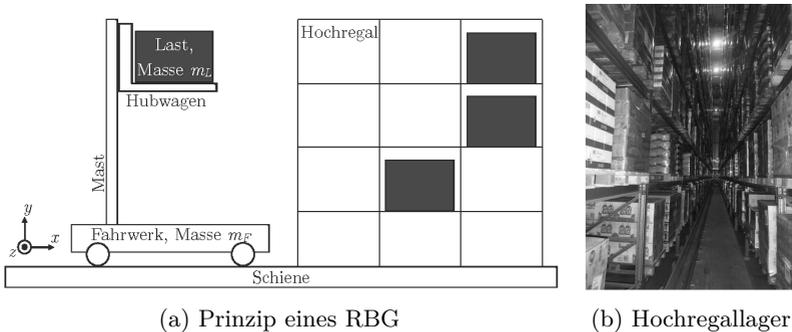


Bild 1: Prinzip eines schienengeführten RBG zur Ein- und Auslagerung von Ladeeinheiten als Last mit der Masse m_L (links), Blick in ein reales Hochregallager im Lebensmitteleinzelhandel (rechts, Nennung Quelle auf Anfrage möglich)

Im Wesentlichen besteht das RBG aus

- einem Fahrwerk für die horizontale Bewegung in der x -Koordinate,
- einer Mastausführung (als Ein- oder Zweimast-RBG),
- einem Hubwerk für die vertikale Bewegung in der y -Koordinate,
- einem Lastaufnahmemittel für die eigentliche Ein- und Auslagerung der Ladeinheit in der z -Koordinate und
- einer Maschinensteuerung [9].

Ladeeinheiten sind beispielsweise Paletten oder Auto-Karosserien mit der Masse m_L . Die Masthöhe ist je nach Anwendung sehr unterschiedlich. Sie kann bei Anwendung in einem Hochregallager zwischen 12 und 50 Metern betragen. Aufgrund von Materialelastizitäten ist der Mast nicht ideal steif. Wegen der Elastizität, der Abmessungen und der Masse des Mastes inkl. der Ladeinheit ist ein RBG ein schwach gedämpftes Feder-Masse-System und somit schwingungsfähig. Das Schwingen des Mastes führt dann zu einer Verlängerung der Spielzeit, was in der Praxis unerwünscht ist.

Die großen Abmessungen des RBG führen aus Sicht des Maschinenbaus und der Regelungstechnik zu sehr schlechten Eigenschaften wie z. B.

- einer hohen bewegten Masse (Hochregallager-RBG haben typischerweise bis 1.500 kg Nutzlast, je nach Höhe des RBG beträgt die Nutzlast lediglich ca. 5 ... 15% der Gesamtmasse),
- einer niedrigen ersten Eigenfrequenz von $f_0 = 0,5 \dots 2$ Hz und
- einem niedrigen Dämpfungsgrad von $D = 0,05 \dots 0,2$.

Die langen Fahrwege und die Forderung nach hohem Durchsatz führen zu bestimmten Fahrgeschwindigkeiten und -beschleunigungen, siehe Tabelle 1. Zusammen mit o.g. Eigenschaften resultiert daraus ein Schwingungspotenzial, was es zu dämpfen gilt, siehe [6].

Tabelle 1: Beispiele typischer, kinematischer Größen von Standard-RBG für Hochregallager, nach [12]. *Die max. Fahrwerk-Beschleunigung ist hier mit $0,5 m/s^2$ niedriger angegeben als in der Literatur, weil in der Praxis oftmals üblich.

Größe	Wertebereich
Max. Geschwindigkeit Fahrwerk	$5 m/s$
Max. Beschleunigung Fahrwerk*	$0,5 m/s^2$
Max. Geschwindigkeit Hubwerk	$2 m/s$
Max. Beschleunigung Hubwerk	$1 m/s^2$

Hersteller von RBG haben unterschiedliche Lösungen für dieses Problem, z. B. Einbau eines Kopfantriebs bzw. Antipendelantrieb an der Mastspitze, Verjüngung des Mastes bzw. Massivbauweise des unteren Mastbereichs oder regelungstechnische Ansätze. Diese Lösungen haben allerdings Nachteile wie z. B. die Notwendigkeit eines zusätzlichen Antriebs, eine höhere Gesamtmasse oder höhere Kosten. Zudem

gibt es in der industriellen Praxis bestehende RBG, insbesondere Paletten-RBG, bei denen nach dem Bremsvorgang Schwingungen auftreten mit der Folge einer Spielzeitverlängerung. Es kann in diesem Fall erst die Ein- oder Auslagerung der Ladeinheit stattfinden, wenn die Schwingung weitestgehend abgeklungen ist. Diese Wartezeit kann mehrere Sekunden, in einem bekannten Fall bis zu 4 Sekunden bei einem Ein- oder Auslagervorgang, dauern. Bei manchen bestehenden Systemen wird diese Wartezeit pauschal eingestellt, da entsprechende Sensoren am Mast zur Messung der Schwingung bzw. aktive regelungstechnische Dämpfungsmaßnahmen nicht vorhanden sind. Beim Paletten-RBG wird zwar i.d.R. ab etwa 20 m Höhe eine Fachfeinpositionierung verwendet. Das bedeutet, die Positionsermittlung in x- und y-Richtung erfolgt dann zusätzlich mittels eines Sensors (meist Kamera oder Lichttaster), um Lageabweichungen des Regalfachs auszugleichen, z. B. aufgrund wechselnder Lasten im Regal sowie Toleranzen aus Fertigung und Montage. Zur Schwingungsdämpfung wird dieses allerdings nicht verwendet, zumal es sich meist nur um Taster handelt. Eine kontinuierliche Wegmessung ist somit nicht möglich, was aber zur aktiven Schwingungsdämpfung notwendig ist. Die Betreiber der Anlagen haben daher großes Interesse durch einfache Methoden nachträglich die Schwingungen zu optimieren. Daher soll der Fokus dieser Benchmark-Anwendung auf einfache nachträglich implementierbare Reglermethoden liegen, z. B. ohne zusätzliche Schwingungssensoren am Mast, als „plug-and-play“-Lösung, um den laufenden Betrieb nicht zu stören.

Labormodell eines Regalbediengeräts

Das Schwingungsverhalten realer RBG lässt sich anhand eines Labormodells nachbilden. Dieses besteht aus einem Fahrwerk, einem biege-elastischen Mast und einer Masse als Last, siehe Bild 2. Ein Encoder erfasst die relative Position des Fahrwerks und ein Dehnungsmessstreifen (DMS) misst die Biegung des Mastes an der untersten Stelle. Die wesentlichen Parameter wie Eigenfrequenz mit $f_0 \approx 0,5 \dots 2$ Hz und Dämpfungsgrad mit $D \approx 0,1$ sind vergleichbar mit einem realen RBG und lassen sich zudem über die Konstruktion variieren, z.B. mit verschiedenen Materialien des biege-elastischen Mastes. Somit lassen sich mit diesem Labormodell grundlegende Schwingungsanalysen durchführen und regelungstechnische Dämpfungsmethoden entwerfen. Gründe für die Verwendung derartiger Modelle sind u.a.

- das sehr anschauliche Testen von regelungstechnischen Methoden (ggü. reinen Computer-Simulationen werden Fehler besser erkannt, daher lassen sich die Ergebnisse besser validieren),

- die Vermeidung potenzieller Schäden an realen Anlagen bei der Implementierung neuer Methoden,
- die praxisnahe Darstellung von Anlagen während Vorlesungen, um viele regelungstechnische Themen besser zu veranschaulichen.

Selbstverständlich ersetzen diese Modelle nicht die Validierung anhand realer Anlagen. Unterschiede zwischen einer realen Anlage und dem Labormodell sind z. B. das Motor-Ansprechverhalten (reale Anlagen haben eine höhere Massenträgheit im Verhältnis zur Motorstärke was den Ruck zusätzlich begrenzt), der Einfluss von Störgrößen (die Energieführungskette beim Labormodell wirkt stärker auf die Bewegung des Fahrwerks ein) oder die Kraftübertragung des Antriebsrads (reale Anlagen sind i.d.R. schlupfbehafet). Darüber hinaus hat die Maschinensteuerung von realen Anlagen typische Funktionen wie eine weitere Ruck-Begrenzung des Motors. Diese Unterschiede bzw. Merkmale sind beim Modell im Rahmen weiterer Entwicklungen zu berücksichtigen.

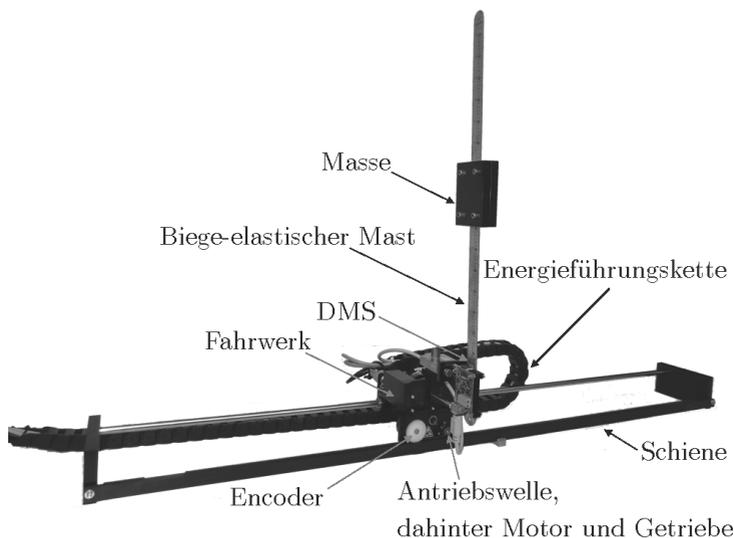


Bild 2: Labormodell eines RBG

Modellbildung und Parameterbestimmung

Zur Modellbildung von RBG existieren mehrere Lösungsansätze, z. B. das Mehrkörpermodell, das Modell mit kontinuierlicher Massenverteilung und Ein- oder Zweimassenmodelle, siehe Bild 3 und vgl. [2, 8]. Darin sind beispielsweise $c_M(y)$, $d_M(y)$, $m_M(y)$ die vertikale Feder-, Dämpfer und Massenverteilung des Mastes bzw. d_φ , c_φ ein Drehdämpfer und eine Drehfeder. Nachfolgend wird ein neues vereinfachtes Modell vorgestellt, um es in weiterführenden Arbeiten mit wenig Aufwand in reale Anlagen in der Praxis zu implementieren, z.B. als online Beobachter-Modell in existierende Maschinensteuerungen.

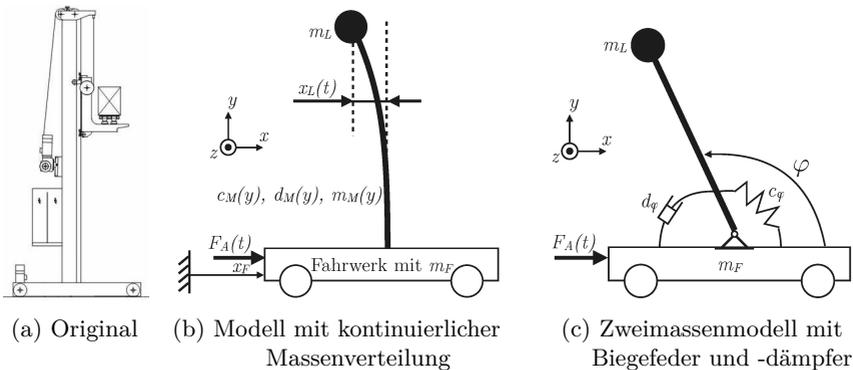


Bild 3: Modellbildung von RBG,

Bild-Quelle 3a: Prof. M. Dietzel, Hochschule Heilbronn

Beim RBG ist i.W. lediglich die Position des Fahrwerks x_F und die Auslenkung am Lastaufnahmemittel x_L von Interesse. Daher wird ein Modell vorgeschlagen, das nur die kartesische raumfeste x_F und lokale x_L Koordinate berücksichtigt, siehe Bild 4.

Der ohnehin kleine Auslenkungswinkel φ wird nicht weiter berücksichtigt. Die Masse m_L beinhaltet sowohl die Nutzlast als auch eine anteilige Masse des Mastes. Das Fahrwerk mit der Masse m_F ist mit der Last m_L über eine Feder c und einem Dämpfer d verbunden, siehe auch Kräftefreischnitt Bild 5.

Darin sind die Kräfte

- der Massenträgheit des Fahrwerks $F_{Fm} = m_F \cdot \ddot{x}_F$,

- der Massenträgheit der Last $F_{Lm} = m_L \cdot (\ddot{x}_F + \ddot{x}_L)$,
- der Feder $F_c = c \cdot x_L$ und
- des Dämpfers $F_d = d \cdot \dot{x}_L$

enthalten. Daraus ergeben sich die Bewegungs-Differentialgleichungen

$$\ddot{x}_F = \frac{1}{m_F} (F_A + d \cdot \dot{x}_L + c \cdot x_L) \quad (1)$$

$$\ddot{x}_L = -\frac{d}{m_L} \cdot \dot{x}_L - \frac{c}{m_L} \cdot x_L - \ddot{x}_F, \quad (2)$$

siehe auch Bild 6. Darin ist zu erkennen, dass die Antriebskraft des Motors die zwei Ausgangsgrößen Position des Fahrwerks x_F und die dazu relative Position des Lastaufnahmemittels x_L beeinflusst.

Die Massen des Fahrzeugs m_F und der Last m_L (jeweils anteilig mit der Mastmasse) als Modell-Parameter sind i.d.R. aus den Datenblättern der Hersteller zu entnehmen. Die Federsteifigkeit c kann über die für RBG typischen Eigenfrequenzen von $f_0 = 0,5 \dots 2$ Hz und mit der Eigenfrequenz von ungedämpften Feder-Masse-Systemen mit $f_0 = \frac{1}{2\pi} \sqrt{\frac{c}{m}}$ bestimmt werden, also mit $c = m_L (f_0 2\pi)^2$.

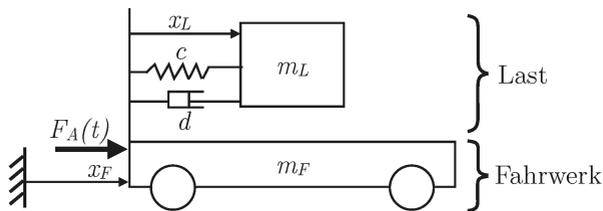


Bild 4: Vereinfachtes Zweimassenmodell eines RBG

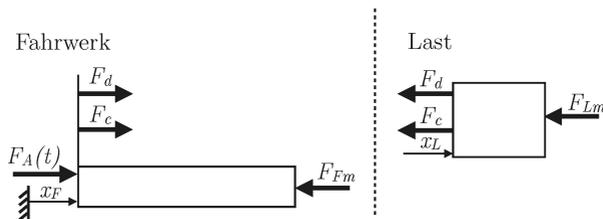


Bild 5: Kräftefreischnitt

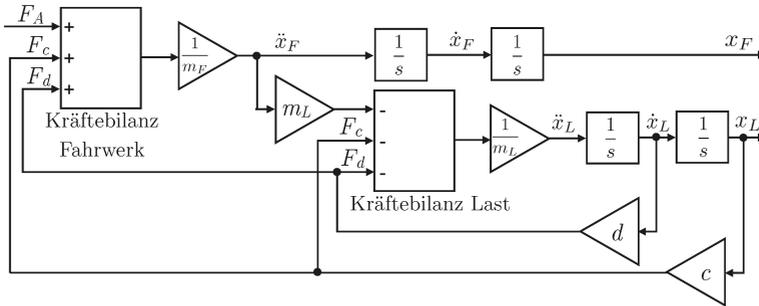


Bild 6: Blockschaltbild der Bewegungs-Differentialgleichungen

Aufgrund des niedrigen Dämpfungsgrades $D = 0,05 \dots 2$ kann der Einfluss auf die tatsächliche Eigenfrequenz $f_e = f_0 \cdot \sqrt{1 - D^2}$ vernachlässigt werden, d.h. $f_e \approx f_0$. Mit dem für ein RBG typischen Dämpfungsgrad mit $D = 0,05 \dots 2$ lässt sich der modellierte mechanische Dämpfer $d = 2 \cdot D \cdot \sqrt{m \cdot c}$ bestimmen. Die Herleitung der genannten Formeln für mechanische Systeme wird in diesem Beitrag nicht weiter vertieft, siehe dazu [8, 11]. Ein Modell für den Motor zur Erzeugung der Antriebskraft F_A wird später kurz beschrieben. Hingegen werden Anregungsfunktionen zur Ruckbegrenzung hier nicht weiter behandelt, siehe dazu [15, 17, 19].

Die in diesem Abschnitt vorgestellte Modellbildung und Parameterbestimmung soll für eine weiterführende Benchmark-Untersuchung zur Auslegung von Reglern auch ohne Labormodell dienen.

Lösungsansätze anhand des Labormodells

In diesem Abschnitt werden verschiedene regelungstechnische Möglichkeiten vorgestellt, um RBG-Schwingungen zu reduzieren. Hierzu existieren einige Lösungen am Beispiel von RBG, siehe [1, 2, 4, 5, 6, 8, 16], aber auch übertragbare Lösungsansätze aus anderen Anwendungen, siehe beispielsweise [14]. Wie bereits erwähnt, kommen diese Lösungen nicht bei allen RBG-Industrieanwendungen zum Einsatz. Die Gründe sind in dieser Arbeit andiskutiert wie z. B. die aufwändige Implementierung oder die fehlende Robustheit, wenn nicht alle Fahr- und Lastsituationen abgedeckt sind.

Als Referenz dient zunächst das Verzögern des Fahrzeugs ohne Berücksichtigung der Mastbewegung, siehe Bild 7. Das Fahrzeug wird geregelt und mittels Sprungfunktion angeregt. Es ist eine maximale Mastauslenkung x_L von ca. $5,5\text{ mm}$ zu beobachten. Weiterhin ist durch die sehr schwache Dämpfung des Labormodells zu sehen, dass die Schwingung nur sehr langsam abklingen. Durch die niedrige Dämpfung des Mastes am Labormodell können nachfolgend die unterschiedlichen Methoden gut verifiziert werden. Diese Referenzmessung spiegelt im übertragenen Sinne die Ausgangslage realer RBG dar, die sich an der Zielposition einschwingen. Konstruktionsbedingt haben reale RBG eine etwas höhere Dämpfung, so dass sie sich schneller einschwingen als bei den folgenden Versuchen exemplarisch am Labormodell gezeigt. Dadurch lassen sich die Methoden besser validieren.

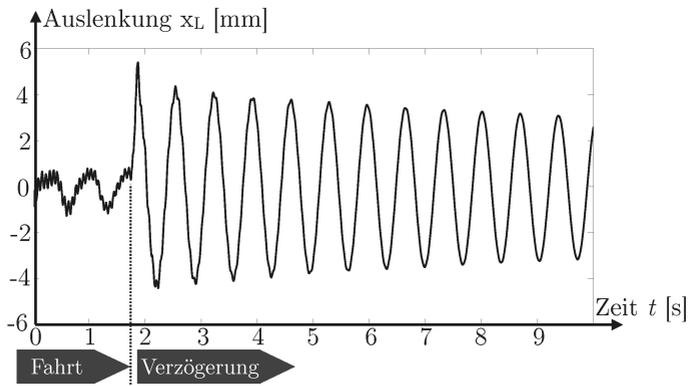


Bild 7: Mastschwingen bei der Verzögerungsphase, wenn das Fahrzeug durch eine Sprungfunktion bis zum Stillstand geregelt wird; das Mastschwingen ist hier regelungstechnisch nicht berücksichtigt

Sollwertanpassung

Eine steuerungstechnische Maßnahme zur Reduzierung von Schwingungen ist die gezielte Vorgabe von Sollwerten. Die Idee hierbei ist, dass Kraftimpulse auf die Anlage einwirken und zwar genau zu den Zeitpunkten, zu denen der Mast durch eine angeregte Schwingung auslenkt, siehe Bild 8. Dies kann über eine Rampe oder über eine doppelte Sprungfunktion als Treppenfunktion erfolgen, siehe z. B. [2, 14, 16]. Bei einer rampenförmigen Anregung ist darauf zu achten, dass die Rampenzeit ein ganzzahliges Vielfaches der Schwingungsperiodendauer $T = \frac{1}{f_0}$ ist. Zu Beginn

der Rampe wird der Mast über die positive Änderung des Eingangssignals angeregt, siehe Zeitpunkt t_0 in Bild 8. Das Rampenende wirkt als Gegenimpuls, aufgrund der negativen Änderung des Eingangssignals, siehe Zeitpunkt t_2 in Bild 8. Ähnlich wirkt eine doppelte Sprungfunktion bzw. Treppenfunktion. Allerdings muss hierbei der zweite Eingangsimpuls zur Hälfte der Mast-Schwingungsperiode erfolgen, weil der Impuls gleichgerichtet mit dem ersten Anregungsimpuls ist, siehe Zeitpunkt t_1 in Bild 8. Diese Methode funktioniert gut bei schwach gedämpften Systemen, was bei einem RBG der Fall ist.

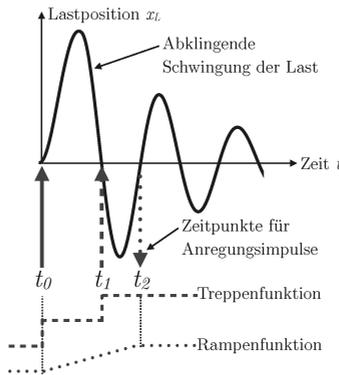


Bild 8: Steuerungsmethoden zur Reduzierung von Schwingungen

Theoretisch kann das Mastschwingen mit dieser Methode deutlich reduziert werden, was zu einer Verbesserung der Schwingungsamplitude und Ausschwingzeit führt, siehe Bild 9. In der Industriepraxis und auch beim Labormodell ist die Verbesserung weniger deutlich ausgeprägt. Der wesentliche Grund ist, dass die Anregung nicht ideal sprung- bzw. rampenförmig erfolgen kann. Dieses ist allerdings notwendig, um damit die besten Ergebnisse zu erzielen. Bei z. B. einer sinusquadrat-förmigen Anregung durchläuft die Anlage über einen längeren Zeitraum Anregungsfrequenzen in der Nähe der Anlagen-Eigenfrequenz, was die Schwingungsamplitude wiederum erhöht. In der Industriepraxis werden ruckartige Anregungen vermieden, nicht zuletzt zur Schonung des Materials. Somit lassen sich die gezeigten guten theoretischen Ergebnisse nicht ganz auf ein reales RBG übertragen. Im Labormodell ist das Verhältnis Motorkraft zur Massenträgheit deutlich höher als in einem realen RBG, so dass die Ergebnisse hier sehr viel besser ausfallen werden (unter Vernachlässigung der hohen mechanischen Beanspruchung bei der Anregung mit der Treppenfunktion).

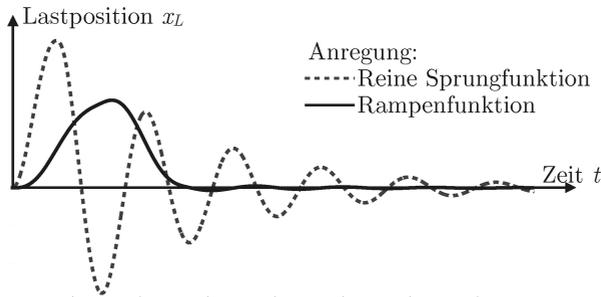


Bild 9: Rechnersimulation der Lastauslenkung mit sprung- und rampenförmiger Kraftanregung zur Verzögerung des modellierten Fahrzeugs

Bei Anregung mit einer Treppenfunktion zeigt das Labormodell im Vergleich zur Referenzmessung eine weniger starke Maximalauslenkung und das Abklingverhalten ist deutlich besser, siehe Bild 10. Allerdings zeigt sich durch die sehr hohe Ruckbelastung die Anregung höherer Mast-Eigenfrequenzen, so dass der Mast in sich schwingt (vergleichbar mit einer „stehenden Welle“). In der Praxis wird die Anregung höherer Eigenfrequenzen zumindest nicht durch den Antrieb hervorgerufen, weil das Kraft-Masse-Verhältnis dazu nicht ausreicht wie hier im Labormodell.

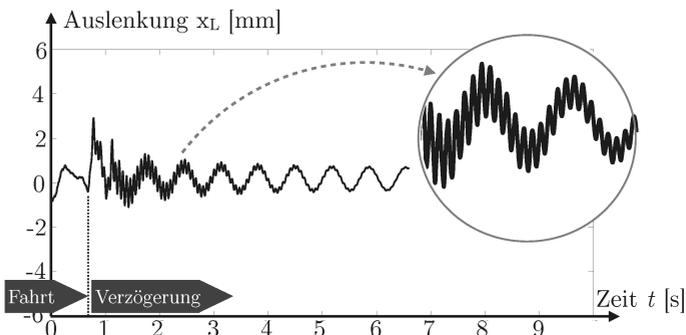


Bild 10: Ergebnis mit Treppenfunktion als Sollwertvorgabe für die Fahrzeug-Regelung; der Mast ist regelungstechnisch nicht berücksichtigt

Der Vorteil dieser Methode ist die sehr einfache Implementierung. Nachteilig ist allerdings, dass die Sprung- bzw. Rampenzeiten auf die Eigenfrequenz der Anlage abgestimmt werden müssen. Die Eigenfrequenz ändern sich mit der Hubhöhe des Lastaufnahmemittels, so dass die Sollwerte ständig angepasst werden müssen (i.d.R. werden die RBG mit wenigen fest eingestellten Standard-Anregungsfunktionen betrieben). Hinzu kommt die bereits geschilderte Ruckbegrenzung realer Anlagen und die fehlende Möglichkeit Störgrößen auszugleichen. Zudem muss der Zeitpunkt der Stützpunkte genau getroffen werden und der Mast darf natürlich im Vorfeld nicht schwingen, sonst überlagern sich die Effekte. Dennoch ist die Sollwertanpassung gut in Kombination mit weiteren Methoden geeignet.

Zustandsregelung mit Schwingungssensor

Das Ziel einer horizontalen RBG-Fahrt ist einerseits eine vorgegebene Fahrzeugposition x_F zu erreichen und andererseits auftretende Schwingungen in der Position des Lastaufnahmemittels x_L bei Erreichung der Fahrzeug-Zielposition zu vermeiden. Regelungstechnisch ist das eine Mehrgrößenregelung und kann beispielsweise mit einer Zustandsregelung gelöst werden [11]. Im Idealfall stehen für alle Größen jeweils ein Sensor-Messsignal bereit. Im RBG-Labormodell misst ein Encoder die Position des Fahrzeugs x_F und ein DMS die Auslenkung der Last x_L . Die Mastschwingung wird dann über die Geschwindigkeit \dot{x}_L bewertet, d.h. sie ist abgeklungen, wenn $\dot{x}_L = 0$ ist. Eine mögliche bleibende Auslenkung in x_L kann über das Fahrzeug ausgeregelt werden. Das DMS-Rohsignal des Labormodells wurde zur Rauschunterdrückung mit einem P-T₁ Filter geglättet und mit einem Gewichtungsfaktor entsprechend der Last-Hubhöhe versehen, weil der DMS am unteren Ende des Mastes befestigt ist. Selbstverständlich können in dieser Anwendung auch Beschleunigungssensoren am Mast angebracht werden, was bei realen RBG u.U. auch vorteilhaft ist, weil dann der Messort direkt an der Lastaufnahme ist. Das Signal muss dann entsprechend integriert werden, um auf die Geschwindigkeit \dot{x}_L zu schließen.

In realen RBG sind üblicherweise Asynchron- bzw. Synchronmotoren eingesetzt. Im Labormodell handelt es sich um ein Gleichstrommotor, der im Folgenden modelliert wird. Der Gleichstrommotor Fahrzeugs erzeugt mit der Eingangsspannung

$$u_e = k \cdot \Phi \cdot \omega + \frac{R}{k \cdot \Phi} M + \frac{L}{k \cdot \Phi} \dot{M} \quad (3)$$

eine Winkelgeschwindigkeit ω und ein Drehmoment M mit dem magnetischen Fluss Φ , dem Widerstand R inkl. sonstiger Reibungen, der Induktivität L und einer Motorkonstanten k , die u.a. Wicklungsanzahl enthält. Auf die Modellbildung von Motoren wird hier nicht näher eingegangen, siehe hierzu [11, 15, 17, 19]. Die

Antriebskraft F_A hängt über den Getriebefaktor i als Hebelarm mit dem Drehmoment $M = F_A \cdot i$ zusammen, ebenso auch die Motor-Winkelgeschwindigkeit ω mit der Fahrzeug-Geschwindigkeit $\dot{x}_F = \omega \cdot i$. Bei Vernachlässigung der Induktivität L bzw. bei konstantem Drehmoment M lässt sich das Motorverhalten durch

$$F_A = -\alpha \cdot \dot{x}_F + \beta \cdot u_e \quad (4)$$

vereinfachen. Die Faktoren α und β sind Abkürzungen und beinhalten den Getriebefaktor i und die Parameter k, Φ, R aus Gleichung (3).

Für den Entwurf des Zustandsreglers werden hier vier Zustandsvariablen definiert, um das System (bzw. das Modell) vollständig zu erfassen, siehe Tabelle 2. Die Struktur dieses Reglers sieht vor, dass die Zustände mit einem Zustandsregler K gewichtet und zurück geführt werden, siehe Bild 11.

Tabelle 2: Zustandsvariablen für den Entwurf des Reglers

Größe bzw. Variable	Formelzeichen
Position Fahrwerk	x_F
Geschwindigkeit Fahrwerk	\dot{x}_F
Position Hubwerk	x_L
Geschwindigkeit Hubwerk	\dot{x}_L

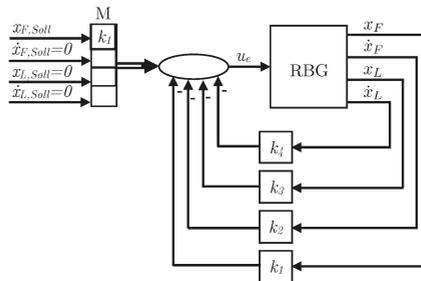


Bild 11: Blockschaltbild der Zustandsregelung des Labormodells

Die Zustandsgleichungen mit der Zustandsdifferential- und Ausgangsgleichung

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot u \\ y &= \mathbf{C}^T \cdot \mathbf{x} \end{aligned} \quad (5)$$

werden mit dem Zustandsvektor $\mathbf{x} = (x_F, \dot{x}_F, x_L, \dot{x}_L)^T$ bzw. dessen Ableitung $\dot{\mathbf{x}} = (\dot{x}_F, \ddot{x}_F, \dot{x}_L, \ddot{x}_L)^T$, der Dynamik- bzw. Systemmatrix

$$\underline{\mathbf{A}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\alpha}{m_F} & \frac{c}{m_F} & \frac{d}{m_F} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{\alpha}{m_L} & \left(-\frac{c}{m_F} - \frac{c}{m_L}\right) & \left(-\frac{d}{m_F} - \frac{d}{m_L}\right) \end{pmatrix}, \quad (6)$$

der Eingangsmatrix $\underline{\mathbf{B}} = \left(0, \frac{\beta}{m_F}, 0, -\frac{\beta}{m_F}\right)^T$, der Ausgangsmatrix $\underline{\mathbf{C}}^T = (1, 0, 1, 0)$ sowie der Eingangsgröße $u = u_e$ und der Ausgangsgröße y unter Berücksichtigung der Gleichungen (1) und (2) beschrieben. Die Ausgangsgröße $y = x_F + x_L$ beschreibt also die Position des Lastaufnahmemittels im Raum zur Erreichung eines bestimmten Regalfaches.

Der Reglerentwurf kann mit einer Polvorgabe erfolgen, hierbei wird die Systemeigenschaft vorgegeben. Für das Labormodell können die Pole mit beispielsweise $\mathbf{P} = (-1, 8 + j1, 9, -1, 8 - j1, 9, -40, -60)$ vorgegeben werden. Zu bemerken ist, dass beim Labormodell die Pole viel weiter links der imaginären Achse liegen dürfen als bei realen Anlagen, weil die mechanische Belastbarkeit beim Labormodell im Größenverhältnis viel höher ist. Gesucht wird nun ein Zustandsregler \mathbf{K} , der das Gesamtsystem so einstellt, dass das Eigenwertproblem $[\lambda_i \mathbf{I} - (\underline{\mathbf{A}} - \underline{\mathbf{B}} \mathbf{K})] \mathbf{x}_i = 0$ mit den vorgegebenen Polen als Eigenwerte löst. Hierzu existieren Algorithmen und mathematische Methoden, die den Zustandsregler berechnen, z.B. der Matlab-Befehl "place(A,B,P)" und auch [11]. Zur Erreichung der stationären Genauigkeit muss noch ein Vorfilter M eingeführt werden, der in diesem Beispiel identisch mit \mathbf{K} ist, siehe Bild 11.

Die Vorteile dieser Methode sind hervorragende Dynamik und Eigenschaft des geregelten Gesamtsystems, siehe Bild 12. Zu erkennen ist die geringe maximale Auslenkung des Mastes x_L sowie die schnell abklingenden Schwingungen. Die im Stillstand noch zu erkennenden Schwingungen mit einer Amplitude von ca. $0,5 \text{ mm}$ sind aufgrund der Beschaffenheit des Labormodells nicht weiter zu reduzieren (u.a. ist das Motoransprechverhalten nicht dazu ausgelegt diese Schwingungen weiter zu reduzieren).

Allerdings ist der hier vorgestellte Zustandsregler auf ein Szenario fest eingestellt, d.h. die im Modell angenommene Federsteifigkeit c und Dämpfung d ist für eine bestimmte Hubhöhe ermittelt worden. Diese Modellvorgaben sind natürlich bei der Berechnung des Zustandsreglers K mit einbezogen. Bei unterschiedlichen Hubhöhen und Lasten muss der Zustandsregler ggf. entsprechend adaptiert werden. Ebenso nachteilig ist, dass der Zustandsregler schwerer interpretierbar ist, als ein klassischer PID-Regler. Das führt zu einer aufwändigen Implementierung

bei einer realen Anlage, da eine heuristische Einstellung und Nachjustierung schwer möglich ist. Zudem ist ein Sensor am Mast erforderlich.

PID-Regler mit Störgrößenaufschaltung

Eine weitere Möglichkeit, um Schwingungen bei der Positionierung des RBG zu reduzieren, ist die Erweiterung des einschleifigen Regelkreises mit einer zusätzlichen Aufschaltung des Steuersignals, siehe Bild 13. Bei genauerer Betrachtung liegen sogar deutliche Parallelen zur Zustandsregelung vor, insbesondere was die Regelung der Lastposition x_L anbelangt. Die Sollposition der Last ist mit $x_{L,Soll} = 0$ vorgegeben.

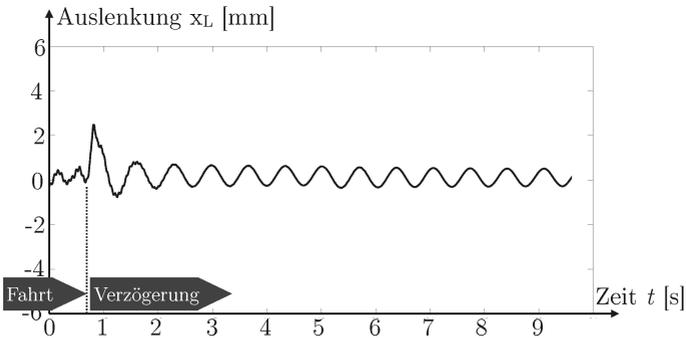


Bild 12: Ergebnis mit Zustandsregelung

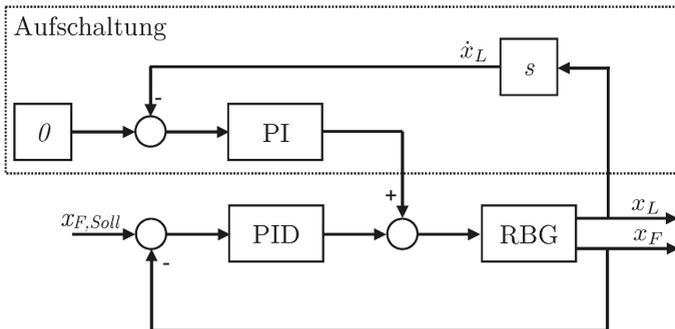


Bild 13: Regelkreisstruktur mit Aufschaltung

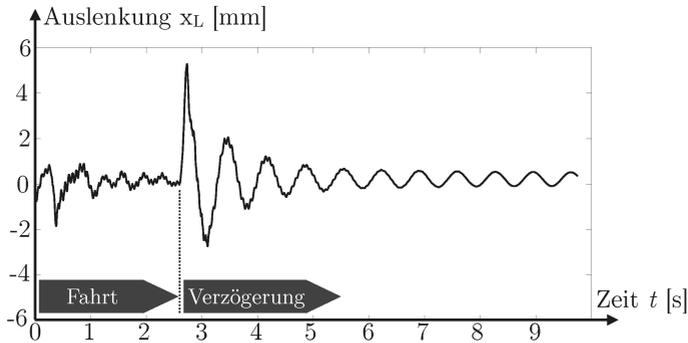


Bild 14: Ergebnis mit PID-Regler und Störgrößenaufschaltung

Für die Anwendung beim Labormodell, liefert diese Methode ähnlich gute Ergebnisse wie der Zustandsregler, siehe Bild 14. Zwar zeigt der hier eingestellte Regler eine noch recht hohe Maximalauslenkung in x_L , allerdings klingen die Schwingungen sehr schnell ab im Vergleich zur Referenzmessung. Durch eine weiter verbesserte Einstellung der Reglerparameter lässt sich dies noch verbessern.

Der wesentliche Vorteil dieser Methode ist die Interpretierbarkeit und die Möglichkeit einer heuristischen Einstellung bei der Inbetriebnahme. Der Nachteil dieser Methode ist die Verwendung eines zusätzlichen Sensors, der am Mast angebracht ist. Des Weiteren müssen die Parameter der Aufschaltung in Abhängigkeit von der Hubhöhe und Last eingestellt werden, um für alle Fahr-Szenarien gute Ergebnisse zu erzielen. Daher bietet diese Lösung erhebliches Potenzial für eine mögliche Implementierung in einer realen Anlage. Allerdings muss die Parameter-Adaption noch ergänzt werden. Hierbei eignet sich wegen der guten Interpretierbarkeit eine Fuzzy-Aufschaltung.

Zusammenfassung

Diese Arbeit stellte ein Labormodell eines Regalbediengeräts (RBG) vor. Anhand des Labormodells wurden Schwingungsanalysen und -dämpfungsmethoden gezeigt. Die Schwingungsdämpfung ist bei realen RBG von großem Interesse, insbesondere bei Paletten-RBG. Hierbei sind Methoden gesucht, die bei existierenden Anlagen nachträglich mit reduziertem Aufwand einsetzbar sind. Anhand dieser industriepraktischen Fragestellung regt diese Arbeit eine Benchmark-Anwendung

an, um geeignete Lösungen zu sammeln und auszuwerten. Die Anwendung mit dem Labormodell kann eine Ergänzung zu bestehenden Benchmark-Problemen sein, siehe [13]. In dieser Arbeit sind bereits Lösungen vorgestellt, die es weiter zu entwickeln gilt, um letztlich in eine vorhandene Maschinensteuerung als „plug-and-play“ Lösung einzubinden. Ein vielversprechender Ansatz auf Basis eines PID-Reglers mit Aufschaltung wurde hier gezeigt. Eine Reduzierung der Mast-schwingung bewirkt eine weitere Spielzeitverkürzung, was für Anlagenbetreiber von großem Interesse ist, siehe auch [18].

Danksagung

Meinen Kollegen Hr. Prof. Michael Dietzel und Hr. Prof. Markus Fittinghoff sowie Hr. Jürgen Melzl und Hr. Prof. Ralf Mikut danke ich herzlich für die stets sehr angenehmen Fachdiskussionen, aus denen letztlich auch die Idee für diesen Beitrag entstand.

Literatur

- [1] D. Arnold und M. Schumacher. „Simulation der Dynamik von Regalbedien-geräten“. In: *Fördern und Heben* 46 Nr. 4. S. 260-265. 1999.
- [2] D. Arnold. „Beeinflussung der Biegeschwingungen des Mastes“. In: *Hebe-zeuge und Fördermittel* 39. S. 576-580. 1999.
- [3] D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier und K. Furmans (Hrsg.). „Handbuch Logistik“. Berlin, Heidelberg: Springer. 2008.
- [4] H. Aschemann, D. Schindele und J. Ritzke. „State and Disturbance Esti-mation for Robust Control of Fast Flexible Rack Feeders“. In: *Modeling, Design, and Simulation of Systems with Uncertainties* Volume 3. S. 333-351. 2013.
- [5] H. Aschemann, G. Kostin, A. Rauh und V. Saurin. „Optimal Control of a Viscoelastic Rack Feeder Based on the Method of Integrodifferential Relations“. In: *Journal of Computer and Systems Sciences International* Volume 54, Issue 2. S. 294-306. 2015.

- [6] M. Bachmeyer, M. Schipplick, T. Thümmel, S. Kessler, W. A. Günther, H. Ulbrich. „Nachschwingungsfreie Positionierung elastischer Roboter durch numerische und analytische Trajektorienplanung am Beispiel Regalbediengerät“. In: *FB 113 Elektrisch-mechanische Antriebssysteme - Innovationen - Trends - Mechatronik*. S. 199-205. 2008.
- [7] DIN 15350 „Regalbediengeräte, Grundsätze für Stahltragwerke, Berechnungen“. Berlin: Beuth. 1992.
- [8] H. Dresig und A. Fidlin. „Schwingungen mechanischer Antriebssysteme“. Berlin, Heidelberg: Springer. 2014.
- [9] Deutsches Nationalkomitee der FEM, VDMA „Terminologie - Regalbediengeräte - Definitionen“. FEM 9.101.
- [10] Deutsches Nationalkomitee der FEM, VDMA „Berechnungsgrundlagen für Regalbediengeräte Tragwerke“. FEM 9.311.
- [11] O. Föllinger. „Regelungstechnik“. Heidelberg: Hüthig. 1994.
- [12] T. Gudehus. „Logistik, Grundlagen – Strategien – Anwendungen“. Berlin, Heidelberg: Springer. 2010.
- [13] F. Hoffmann, R. Mikut, A. Kroll, M. Reischl, O. Nelles, H. Schulte und T. Bertram. „Computational Intelligence: State-of-the-Art Methoden und Benchmarkprobleme“. In: *Proc., 22. Workshop Computational Intelligence*. S. 15-29. Karlsruhe: KIT Scientific Publishing 2012.
- [14] P. Hubinský und T. Pospiech. „Slosh-Free Positioning of Containers with Liquids and Flexible Conveyor Belt“. In: *Electrical Engineering* Volume 61, Number 2. S. 65–74. 2010.
- [15] R. Mansius. „Praxishandbuch Antriebsauslegung“. Würzburg: Vogel. 2012.
- [16] M. Schipplick und W. A. Günthner. „Potentiale einer flachheitsbasierten Antriebssteuerung am Beispiel Regalbediengerät“. In: *at - Automatisierungstechnik* Band 59, Heft 4. S. 242–247. 2011.
- [17] D. Schröder. „Elektrische Antriebe - Regelung von Antriebssystemen“. Berlin, Heidelberg: Springer. 2014.
- [18] S. Seemüller. „Durchsatzberechnung automatischer Kleinteilelager im Umfeld des elektronischen Handels“. München: Herbert Utz Verlag. 2005.
- [19] J. Weidauer. „Elektrische Antriebstechnik“. Erlangen: Publicis Publishing. 2013.

Zur Homogenisierung von Breitbandtestsignalen für die nichtlineare Systemidentifikation am Beispiel eines elektromechanischen Stellantriebs

Matthias Gringard, Andreas Kroll

Universität Kassel, Fachbereich Maschinenbau,
Fachgebiet Mess- und Regelungstechnik
E-Mail: {matthias.gringard;andreas.kroll}@mrt.uni-kassel.de

Einleitung

Aufgrund der stetig zunehmenden Komplexität dynamischer technischer Systeme und des daraus resultierenden ebenfalls zunehmenden Aufwands physikalischer Modellbildung findet die datengetriebene Modellbildung immer mehr Anwendung. Da ein identifiziertes Modell nur das dynamische Verhalten sicher wiedergeben kann, welches in den Identifikationsdaten enthalten ist, ist es von großem Interesse, das zugrunde liegende System so anzuregen, dass möglichst das gesamte dynamische Verhalten durch die zugehörigen Ausgangsdaten abgebildet wird. Zu diesem Zweck sollen Anregungs- (oder Test-)signale gezielt entworfen werden, um Datensätze mit einem hohen Informationsgehalt zu erzeugen. Es werden prozessmodellfreie und prozessmodellbasierte Verfahren zum Testsignaldesign unterschieden, wobei berücksichtigt werden muss, dass die Modellstruktur, welche zur Durchführung der modellbasierten Methoden benötigt wird, im Allgemeinen zu Beginn unbekannt und selbst Ziel der Identifikation ist. So kann eine optimale Versuchsplanung (ODE) wie in [7] besprochen nur bei bekannter Modellstruktur durchgeführt werden.

Aus diesem Grund soll untersucht werden, ob prozessmodellfreie Entwurfsverfahren, welche ohne Informationen über die Modellstruktur auskommen, zu besseren Modellen führen als Standardbreitbandtestsignale. Hierbei wird die Klasse der nichtlinearen dynamischen lokal affinen Takagi-Sugeno(TS-)Modelle betrachtet. Analog zur modellfreien Versuchsplanung für eine statische nichtlineare Modellbildung [8] zielen die modellfreien Verfahren für dynamische nichtlineare Modelle darauf ab, eine möglichst gleichmäßige Verteilung der Daten im Eingangsgrößenraum zu erzeugen. Es soll vermieden werden, einzelne Bereiche stärker anzuregen, sofern kein Wissen über das System vorhanden ist. Im Gegensatz zu statischen Systemen ist es bei dynamischen Systemen jedoch nicht möglich, die Designpunkte

der Eingangsgröße unabhängig voneinander zu platzieren, da sie eine Trajektorie im Eingangsgrößenraum repräsentieren. Ebenso sind zur Bewertung der Datenabdeckung die vergangenen Werte der Ausgangsgröße zu berücksichtigen.

Im vorliegenden Beitrag werden Breitbandtestsignale, welche zur Identifikation linearer dynamischer Systeme entwickelt wurden [6] zur Identifikation nichtlinearer dynamischer TS-Systeme verwendet. Die Signale werden zusätzlich so angepasst, dass der Wertebereich gleichmäßig abgedeckt wird, um so Ideen der modellfreien Versuchsplanung für statische Modelle aufzugreifen. Dazu wird zunächst die Homogenisierungsmethode aus [5] als Beispiel eines prozessmodellfreien Verfahrens zum Testsignaldesign untersucht. Die eindimensionale Homogenisierung (ODH) basiert darauf, iterativ die Phasen der diskreten Fourier-Transformation (DFT) eines Signals so anzupassen, dass der Wertebereich des homogenisierten Signals gleichmäßig abgedeckt wird. Insbesondere eignet sich diese Methode für Multisinus-Signale, da das Amplitudenspektrum bei der Homogenisierung erhalten bleibt und dadurch Designparameter unabhängig von dieser Methode vorgegeben werden können.

Nach einer Einführung in die Methode erfolgt in diesem Beitrag die Identifikation eines elektromechanischen Stellantriebs [3] und die Analyse der Ergebnisse zwecks Demonstration der erreichbaren Ergebnisse.

Testsignaldesign und Identifikation

Dieser Abschnitt beginnt mit der Vorstellung der verwendeten Modellklasse und ihrer Nomenklatur. Im Anschluss daran werden das Identifikationsverfahren und der Testsignalentwurf behandelt. Der Diskussion der Fallstudie wird die Einführung der Bewertungskriterien vorgezogen.

Verwendete Modellklasse

Bei der verwendeten Modellklasse handelt es sich um linear affine Takagi-Sugeno-(TS-)Fuzzy-Modelle, wobei die lokalen Teilmodelle eine homogene Struktur haben. TS-Modelle basieren darauf, lokale Teilmodelle gewichtet zu überlagern. Da diese linear affin sind, wird das nichtlineare Verhalten des TS-Modells ausschließlich über die Partitionierung eingestellt. Das i -te lokale Teilmodell ist dann in folgender Form darstellbar:

$$\hat{y}_i(k) = - \sum_{j=1}^{n_{yc}} a_{i,j} \cdot y(k-j) + \sum_{l=1}^{n_{uc}} b_{i,l} \cdot u(k-l) + a_{i,0} \quad (1)$$

Hierbei sind $\hat{y}_i(k)$ der Modellausgang des i -ten lokalen Teilmodells, $a_{i,j}$ und $b_{i,l}$ die Koeffizienten der Ein- und Ausgangsgrößen beziehungsweise ihrer Verzögerungen. n_{yc} und n_{uc} sind die maximalen Verzögerungen in der Ein- und Ausgangsgröße, k ist die diskrete Zeit und $a_{i,0}$ der affine Term. Um die Parameterschätzung vorzubereiten, wird die Differenzgleichung (1) noch umgeformt:

$$\begin{aligned}
 \hat{y}_i(k) &= - \underbrace{\begin{bmatrix} y(k-1) \\ \vdots \\ y(k-n_{yc}) \end{bmatrix}}_{\varphi_y^T(k-1)} \underbrace{\begin{bmatrix} a_{i,1} \\ \vdots \\ a_{i,n_{yc}} \end{bmatrix}}_{\Theta_{i,y}} + \underbrace{\begin{bmatrix} u(k-1) \\ \vdots \\ u(k-n_{uc}) \end{bmatrix}}_{\varphi_u^T(k-1)} \underbrace{\begin{bmatrix} b_{i,1} \\ \vdots \\ b_{i,n_{uc}} \end{bmatrix}}_{\Theta_{i,u}} + a_{i,0} \\
 &= \underbrace{\begin{bmatrix} 1 & \varphi_y^T(k-1) & \varphi_u^T(k-1) \end{bmatrix}}_{\varphi^T(k-1)} \underbrace{\begin{bmatrix} a_{i,0} \\ \Theta_{i,y} \\ \Theta_{i,u} \end{bmatrix}}_{\Theta_i} = \varphi^T(k-1) \cdot \Theta_i \quad (2)
 \end{aligned}$$

Hierbei sind die $\varphi^T(k-1)$ die Regressions- und die Θ_i die Parametervektoren. Falls eine Totzeit auftritt, kann diese direkt im u -Term berücksichtigt werden. Das Gesamtmodell ergibt sich aus den mit Zugehörigkeitsfunktionen $\mu_i(\mathbf{z}(k-1))$ gewichteten Teilmodellen. Um zu verdeutlichen, dass der Regressionsvektor und die Scheduling-Variable $\mathbf{z}(k-1)$ vor allem von den vergangenen Werten der Ein- und Ausgangsgröße bis zum Zeitpunkt $k-1$ abhängen, wird ihnen das Argument $(k-1)$ gegeben. Die Zugehörigkeitsfunktionen werden entsprechend des Fuzzy C-Means (FCM) Algorithmus angesetzt:

$$\mu_i(\mathbf{z}(k-1)) = \left[\sum_{j=1}^c \left(\frac{\|\mathbf{z}(k-1) - \mathbf{v}_i\|_2}{\|\mathbf{z}(k-1) - \mathbf{v}_j\|_2} \right)^{\frac{2}{\nu-1}} \right]^{-1} \quad (3)$$

Die μ_i sind Funktionen der Scheduling-Variablen $\mathbf{z}(k-1)$. Häufig ist die Scheduling-Variable eine Funktion des Regressionsvektors, kann aber auch etwas vollständig anderes sein, wenn auf einem anderen Weg Wissen in den Modellierungsprozess eingebracht werden konnte. c ist die Anzahl der lokalen Modelle, die \mathbf{v}_i sind Prototypen der Partitionen und $\nu \in \mathbb{R}^{>1}$ der Unschärfeparameter. Die Zugehörigkeitsfunktionen, welche im FCM-Algorithmus verwendet werden, haben den Vorteil, dass sie die Orthogonalitätsbedingung erfüllen:

$$\sum_{i=1}^c \mu_i(\mathbf{z}(k-1)) = 1 \quad (4)$$

Somit kann das Gesamtmodell nun angegeben werden als:

$$\begin{aligned}
 \hat{y}(k|k-1) &= \frac{\sum_{i=1}^c \mu_i(\mathbf{z}(k-1)) \cdot \boldsymbol{\varphi}^T(k-1) \cdot \boldsymbol{\Theta}_i}{\sum_{i=1}^c \mu_i(\mathbf{z}(k-1))} \\
 &= \sum_{i=1}^c \phi_i(\mathbf{z}(k-1)) \cdot \boldsymbol{\varphi}^T(k-1) \cdot \boldsymbol{\Theta}_i \\
 &= \sum_{i=1}^c \mu_i(\mathbf{z}(k-1)) \cdot \boldsymbol{\varphi}^T(k-1) \cdot \boldsymbol{\Theta}_i \tag{5}
 \end{aligned}$$

Die letzte Zeile in (5) gilt wegen (4).

Identifikationsverfahren

Mit den Abkürzungen $\mu_i(\mathbf{z}(k-1)) = \mu_{ik}$ und $\boldsymbol{\varphi}^T(k-1) = \boldsymbol{\varphi}_k^T$ folgt aus (5):

$$\hat{y}(k|k-1) = \begin{bmatrix} \mu_{1k} \boldsymbol{\varphi}_k^T & \cdots & \mu_{ck} \boldsymbol{\varphi}_k^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\Theta}_1 \\ \vdots \\ \boldsymbol{\Theta}_c \end{bmatrix} \tag{6}$$

Werden N Beobachtungen berücksichtigt, kann (6) $(N - n_{yc})$ mal mit den zur Verfügung stehenden Beobachtungen ausgewertet werden. Es folgt:

$$\hat{\mathbf{y}} = \boldsymbol{\Phi}_E \boldsymbol{\Theta} \tag{7}$$

$\boldsymbol{\Phi}_E$ ist die erweiterte Regressionsmatrix, $\boldsymbol{\Theta}$ der Vektor aller lokalen Modellparameter und $\hat{\mathbf{y}}$ der Regressand, welcher $(N - n_{yc})$ Beobachtungen enthält. Hierbei handelt es sich um ein Schätzproblem, welches linear in den Parametern ist, sofern die \mathbf{v}_i bekannt sind. Daher kann bei Verwendung eines quadratischen Gütekriteriums die Lösung mittels der Methode der kleinsten Quadrate als geschlossene Formel angegeben werden zu:

$$\hat{\boldsymbol{\Theta}} = (\boldsymbol{\Phi}_E^T \boldsymbol{\Phi}_E)^{-1} \boldsymbol{\Phi}_E^T \hat{\mathbf{y}} \tag{8}$$

Die Lösung 8 liefert die optimale Lösung für die Einschrittprädiktion (one-step-ahead-prediction (OSAP)). Um sie berechnen zu können, müssen sowohl die Anzahl der Teilmodelle c als auch die Prototypen der Teilmodelle \mathbf{v}_i bekannt sein. Damit sind wichtige strukturelle Entscheidungen, wie die Festlegung der

Scheduling-Variable, der Anzahl der Teilmodelle sowie der Zugehörigkeitsfunktionen zu treffen, wenn eine Schätzung der lokalen Modellparameter nach (8) erfolgen soll. Zur Bestimmung der Prototypen \mathbf{v}_i wird der FCM-Algorithmus genutzt. Dabei ist es wichtig zu beachten, dass die so ermittelten Prototypen optimal bezüglich des Gütekriteriums des FCM-Algorithmus sind, was im Allgemeinen jedoch nicht bedeutet, dass sie bezüglich der Prädiktionseigenschaften des Modells optimal sind.

Der mit (8) bestimmte Parametervektor $\hat{\Theta}$ minimiert die Summe der quadratischen Abweichungen der Einschrittprädiktion bei vorgegebenen Prototypen \mathbf{v}_i . Von größerem Interesse ist die Simulationsfähigkeit des Modells. So werden in einer Simulation nicht gemessene Werte zur Vorhersage des nächsten Werts der Ausgangsgröße genutzt, sondern die zurückliegenden Modellausgänge. Es handelt sich also um eine rekursive Auswertung. Es ist offensichtlich, dass die Güte der Simulation ein härteres Kriterium ist als die Einschrittprädiktion. Aus diesem Grund werden die Parameter mit einem nichtlinearen Optimierungsalgorithmus bezüglich des rekursiv ausgewerteten Modells optimiert. Die Parameter eines lokalen Teilmodells Θ_i und der entsprechende Prototyp \mathbf{v}_i werden zu einem Gesamtparametervektor

$$\Theta_{\text{ges},i}^T = [\Theta_i^T \quad \mathbf{v}_i^T] \quad (9)$$

des Teilmodells zusammengefasst, so dass in der nichtlinearen Optimierung die Partitionierung und die lokalen Modelle parallel optimiert werden. Wird ein quadratisches Gütekriterium benutzt, so gilt:

$$\hat{\Theta}_{\text{ges}} = \underset{\Theta_{\text{ges}}}{\operatorname{argmin}} \sum_{k=1}^N (y_k - \hat{y}_k((\Theta_{\text{ges}})))^2 \quad (10)$$

mit

$$\Theta_{\text{ges}}^T = [\Theta_{\text{ges},1}^T \quad \cdots \quad \Theta_{\text{ges},c}^T] \quad (11)$$

Das Optimierungsproblem wird mit der MATLAB-Funktion `lsqnonlin` gelöst, welche standardmäßig einen Trust-Region-Reflective-Algorithmus nutzt. Die Optimierung wird mit den Modellparametern aus (8) und den Prototypen aus dem FCM-Algorithmus initialisiert.

Testsignalentwurf

Die Testsignale werden mit der Homogenisierungsmethode aus [5] entworfen. Das Ziel der Methode ist es, eine Gleichverteilung im Eingangsgrößenraum herzustellen.

Dazu wird der Eingangsgrößenraum in n_f gleich große Intervalle zerlegt. Die tatsächliche Verteilung wird dann durch die kumulierte quadratische Abweichung der Daten in einem Intervall n_i zur idealen Anzahl der Daten bei Gleichverteilung pro Intervall n_{ideal} bewertet. Die ideal erwartete Anzahl der Daten pro Intervall n_{ideal} ergibt sich aus dem Quotienten der Anzahl N der Werte und der Anzahl der Intervalle n_f :

$$n_{\text{ideal}} = \frac{N}{n_f} \quad (12)$$

Als Maß für die Abdeckung wird der Coverage-Index J_{CI} verwendet:

$$J_{\text{CI}} = \sum_{i=1}^{n_f} (n_i - n_{\text{ideal}})^2 \quad (13)$$

In der Literatur werden dem Index J_{CI} Faktoren hinzugefügt, welche jedoch nur den Zweck erfüllen, den Index numerisch klein zu halten. Eine richtige Normierung, welche zur Folge hat, dass der Index bei verschiedenen Intervallanzahlen n_f und Beobachtungen N gleich bleibt, so dass sich die Abdeckungen verschiedener Versuche vergleichen lassen, ist durch das Hinzufügen der Faktoren aus der Literatur nicht gegeben, weshalb an dieser Stelle auf sie verzichtet wurde. Aus demselben Grund kann als Abbruchkriterium kein fester Wert für J_{CI} genutzt werden und als Abbruchkriterium wird eine Mindeständerung von J_{CI} eingeführt:

$$\Delta J_{\text{CI}} = J_{\text{CI},i+1} - J_{\text{CI},i} \leq \epsilon \quad (14)$$

Das Kriterium ist konvergenzorientiert. Es besteht die Möglichkeit, das Kriterium um eine kostenorientierte Komponente zu erweitern. Die Schranke ϵ muss vorgegeben werden. Ein Iterationsschritt der Homogenisierungsmethode ist dann:

1. Bestimmung von J_{CI} für ein Signal $[u_k] = \mathbf{u}$ und Check des Abbruchkriteriums (14).
2. Sortierung der Zeitreihe \mathbf{u} der Größe nach. Die Sortierung kann formal mit einer Permutationsmatrix \mathbf{T} ausgedrückt werden: $\mathbf{u}' = \mathbf{T}\mathbf{u}$
3. Definition einer Zeitreihe \mathbf{u}'' , welche eine Gerade ist.
4. Rücksortierung der Zeitreihe \mathbf{u}'' mit $\mathbf{q} = \mathbf{T}^{-1}\mathbf{u}''$
5. Berechnung der zeitdiskreten Fouriertransformierten (DFT)
 $\mathbf{Q} = \text{DFT}(\mathbf{q})$ und $\mathbf{U} = \text{DFT}(\mathbf{u})$
6. Erzeugung einer neuen DFT $\mathbf{R} = [R_k]$, $R_k = |U_k| \cdot \exp(j \cdot \arg(Q_k))$

7. Anwendung der inversen zeitdiskreten Fouriertransformation (IDFT) auf \mathbf{R} , $\mathbf{r} = \text{IDFT}(\mathbf{R})$
8. Setze $\mathbf{u} = \mathbf{r}$

Im 6. Schritt wird aus den beiden Fouriertransformierten \mathbf{U} und \mathbf{Q} ein neue Fouriertransformierte \mathbf{R} erzeugt, indem die Beträge von \mathbf{U} und die Phasen von \mathbf{Q} kombiniert werden.

Diskussion der Wirkungsweise der Methode

Mit der Homogenisierungsmethode werden die Phasen der zeitdiskreten Fouriertransformierten eines Signals iterativ so angepasst, dass sich im Zeitbereich eine möglichst gleichmäßige Verteilung der Daten einstellt. Da nur die Phasen geändert werden, bleibt, wie auch schon im vorangegangenen Abschnitt erwähnt, das Amplitudenspektrum des Signals erhalten. Die Signale, welche mit dieser Methode entworfen wurden, sollen ohne Vorwissen über die Systemdynamik zu erfordern zu einer besseren Anregung des Systems führen, was wiederum zu einer verbesserten Identifikation führen soll. Zur Diskussion der Wirkung der Anregung werden zwei Aspekte betrachtet, das Amplitudenspektrum des Ausgangssignals sowie seine Verteilung. Da die meisten technischen Systeme Tiefpasscharakter haben, sollen die Effekte am transparenten Beispiel eines linearen Filters 1. Ordnung mit der Übertragungsfunktion

$$G_{\text{PT1}}(s) = \frac{1}{\frac{2}{\pi}s + 1} \quad (15)$$

untersucht werden. Dazu wird ein Multisinus-Signal verwendet:

$$u(t) = \sum_{i=1}^n \sin(2\pi i f_0 t + \phi_i) \quad (16)$$

Hierbei ist $n = 250$ die Anzahl der Harmonischen, $f_0 = 0,01$ Hz die Frequenz der Grundharmonischen und die Phasen ϕ_i werden als Schroederphasen angesetzt:

$$\phi_i = -\frac{\pi \cdot i \cdot (i - 1)}{n} \quad (17)$$

Bild 1 zeigt Systemein- und ausgangssignale des linearen Filters aus (15). Auch wenn es aufgrund des dargestellten Ausgangssignals in Bild 1 (unten links) anders erscheint, hat die stationäre Systemantwort des Filters auf die gegebene Anregung einen Mittelwert bei Null. Eine Abweichung ergibt sich nur durch die Überlagerung des Einschwingverhaltens. Durch das Filter wird das Amplitudenspektrum

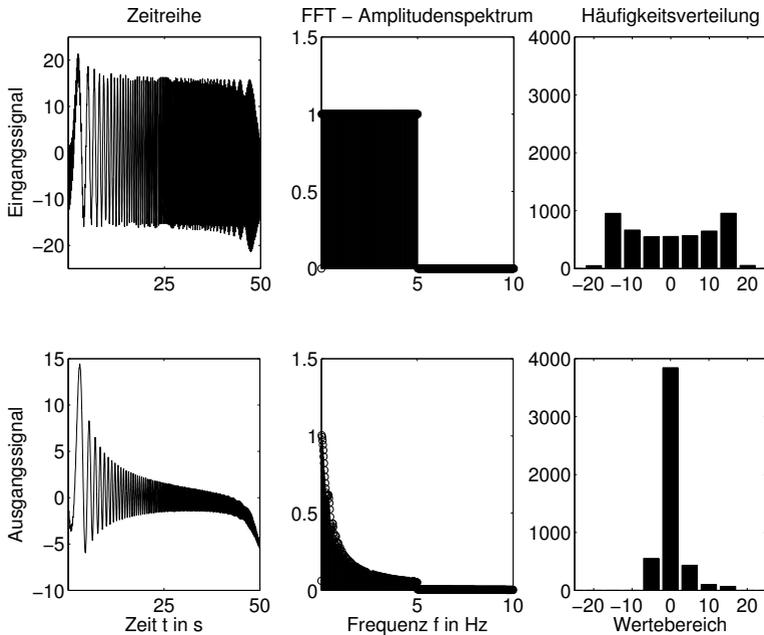


Bild 1: Ein- (oben) und Ausgangssignale (unten) eines linearen Filters bei Multisinus-Testsignal

verzerrt, wie Bild 1 (unten Mitte) zeigt. So werden die Amplituden, welche zu den höheren Frequenzen des Anregungsspektrums gehören, erwartungsgemäß immer weiter abgeschwächt. Die Abschwächung eines Großteils der auftretenden Amplituden hat im Zeitsignal den Effekt, dass eine größere Häufung der Daten um den Gleichanteil zu finden ist, wie in Bild 1 (unten links, unten rechts). Dazu wird folgende Arbeitshypothese aufgestellt: Die Fourier-Analyse in Bild 2 wurde mit dem in (16) definierten Multisinus-Signal am künstlichen System aus [6, 9] durchgeführt³:

$$y(k+1) = \frac{y(k) \cdot y(k-1) \cdot (y(k) + 2,5)}{1 + y^2(k) + y^2(k-1)} + u(k) \quad (18)$$

Die Systemantwort eines nichtlinearen Systems auf eine Summe harmonischer Schwingungen (Multisinus) besteht nach dem Abklingen der stabilen Eigenbewe-

³ In (18) fehlt in [6] ein Additionszeichen im Vergleich zur Originalquelle [9].

gungen aus Oszillationen, welche sich nach einer Fourier-Analyse hauptsächlich aus den Frequenzanteilen des Multisinus zusammensetzt, jedoch auch weitere Frequenzanteile, welche nicht im Eingangsspektrum vorhanden waren, enthält, wie Bild 2 vermuten lässt. So ist auch bei nichtlinearen Systemen zu erwarten, dass

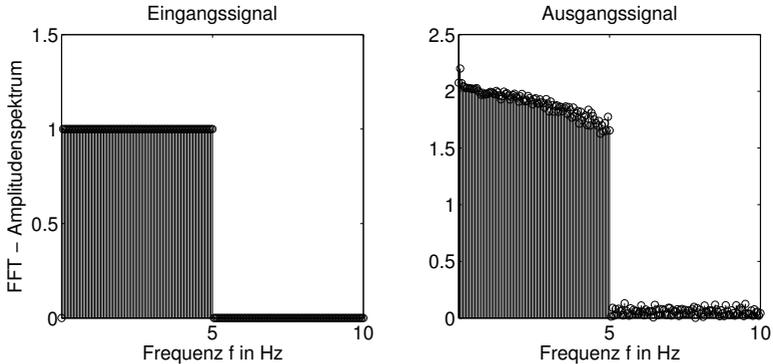


Bild 2: Fourier-Analyse der Antwort des nichtlinearen Systems (18) auf Testsignal (16)

durch den Filtereffekt eines realen technischen Systems höherfrequente Anteile in der Systemantwort eine geringere Amplitude aufweisen. Wird ein nichtlineares System breitbandig angeregt, liefern die schwach gefilterten Frequenzanteile bei einer Parameterschätzung einen größeren Beitrag zum Modellfehler. Somit führt eine Homogenisierung des Eingangssignals nicht zu einer Homogenisierung des Ausgangssignals, welches allerdings oft den Scheduling-Raum mit aufspannt.

Bewertungskriterien für die Modellgüte

Die Modellgüte soll anhand zweier Kriterien bewertet werden. Als Vertreter eines mittelnden Maßes wird der **root mean square error** (RMSE) (19) verwendet. Als Vertreter eines lokalen Maßes wird der **maximum absolute error** (MAE) (20) verwendet. Die beiden Maße berechnen sich wie folgt:

$$J_{\text{RMSE}} = \sqrt{\frac{1}{N} \sum_{k=1}^N (y(k) - \hat{y}(k))^2} \quad (19)$$

$$J_{\text{MAE}} = \max_{1 \leq k \leq N} |y(k) - \hat{y}(k)| \quad (20)$$

Es ist dabei wichtig zu berücksichtigen, dass Modellgüte bezüglich eines Validierungsdatensatzes höher als die bezüglich des Identifikationsdatensatzes zu bewerten ist. Weitere verbreitete Kriterien zur Modellbewertung können [6] entnommen werden.

Fallstudie

Die vorgestellte Methode wird anhand der in [3] als Benchmarkproblem für die datengetriebene Modellbildung vorgestellte elektromechanische Drosselklappe demonstriert. Aus Vergleichszwecken wird das in [3] verwendete Identifikations- und Validierungssignal benutzt.

Elektromechanische Drosselklappe

Die Drosselklappe besteht aus einem Gleichstrommotor, einem Getriebe, einer Rückstellfeder und der Drosselscheibe selbst, welche in einem Metallgehäuse verbaut ist. Mit einem Potentionmeter wird die Winkelstellung der Platte, welche die Ausgangsgröße $y(k)$ des Systems ist, gemessen. Die Winkelstellung kann Werte zwischen 10° , was der vollständig geschlossenen Klappe entspricht, und 90° , was der vollständig geöffneten Klappe entspricht, annehmen. Der Motor wird mit einem pulsweitenmodulierten Ansteuersignal versorgt. Sein Tastverhältnis (in %) ist die Systemeingangsgröße $u(k)$. Da das System größtenteils aus elementaren Komponenten besteht, könnte angenommen werden, dass sich auf einfache Art und Weise ein theoretisches Modell herleiten lässt. Jedoch kann mittels theoretischer Modellbildung die zustandsabhängige Reibung sowie das nichtlineare Verhalten der Feder nur aufwändig abgebildet werden [1]. Auch sind Größen, welche bei der theoretischen Modellbildung häufig als bekannt oder wenigstens einfach bestimmbar vorausgesetzt werden, im Allgemeinen unbekannt. Aus diesem Grund wird die datengetriebene Modellbildung eingesetzt, um ein geeignetes Simulationsmodell für die elektromechanische Drosselklappe zu erzeugen.

Vorbereitung der Identifikation

Der Regressionsvektor wird entsprechend [6, 3] gewählt als:

$$\varphi_k^T = [u(k-1) \quad y(k-1) \quad y(k-2)] \quad (21)$$

Die Schedulingvariable wird aus dem Regressionsvektor abgeleitet als:

$$\mathbf{z}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \end{bmatrix} \boldsymbol{\varphi}_k = [u(k-1) \quad (y(k-2) - y(k-1))]^T \quad (22)$$

Für die Identifikation wurden $c = 8$ Teilmodelle sowie ein Unschärfeparameter von $\nu = 1,1$ genutzt. Die Anzahl der Teilmodelle wurde in [1] mittels Clustervaliditätsmaßen ermittelt und hier zwecks Vergleichbarkeit der Ergebnisse übernommen. Eine passende Abtastzeit wurde in [1] experimentell zu $T_A = 10^{-2}$ s ermittelt.

Entwurf der Testsignale für die Fallstudie

Zur Identifikation der Drosselklappe werden Multisinus-Testsignale entworfen:

$$u_k = \sum_{i=1}^{n_{ms}} A_i \cdot \sin(2\pi f_i k T_A + \phi_i) \quad (23)$$

Hierbei ist n_{ms} die Anzahl der verwendeten Harmonischen. Eine geeignete Bandbreite des Signals wurde in [1] auf 5 Hz festgelegt. Die obere Grenzfrequenz des Multisinus-Signals wird damit als $f_{max} = 5$ Hz gewählt. Mit der gewählten Signaldauer von $T_{max} = 100$ s werden die Frequenzanteile f_i bestimmt:

$$f_i = h_i \cdot \Delta f = \frac{h_i}{T_{max}} \quad (24)$$

Hierbei sind die h_i die ausgewählten Frequenzanteile aus der Anzahl der möglichen $f_{max}/\Delta f$ Harmonischen:

$$h_i \in \mathbb{H} \\ \mathbb{H} = \left\{ x \mid x \in \mathbb{N}, \frac{f_{max}}{\Delta f} \in \mathbb{R} \wedge x \in \left[1, \frac{f_{max}}{\Delta f} \right] \right\} \quad (25)$$

Um unerwünschte nichtlineare Effekte zu vermeiden, wurden nur primharmonische Frequenzanteile genutzt [1]. Für das Basis-Signal werden Schroederphasen verwendet. Da nicht alle Höherharmonischen der Grundfrequenz genutzt werden, muss die Formel zur Bestimmung der Schroederphasen (17) noch angepasst werden:

$$\phi_i = -\frac{\pi \cdot h_i \cdot (h_i - 1)}{n_{ms}}, \quad i = 1 \dots n_{ms} \quad (26)$$

Alle Amplituden A_i in (23) werden zu $A_i = 1$ gesetzt. Für das sich so ergebene Multisinus-Signal wird die Homogenisierung durchgeführt. Da bei der Homogenisierung das Amplitudenspektrum unberührt bleibt und eine Gleichverteilung der Datenpunkte erreicht werden soll, schrumpft der Wertebereich des homogenisierten Signals. Aus diesem Grund wird das homogenisierte Multisinus-Signal auf den Wertebereich des ursprünglichen Signals skaliert.

Die in [1] aufgeführten Daten wurden mit einer anderen Drosselklappe, welche nicht mehr zur Verfügung steht, erzeugt. Um den Austausch des Prüflings zu berücksichtigen, wurden nicht nur für die neuen Signale, sondern auch für das Multisinus-Signal mit Schroederphasen aus [1] Versuche durchgeführt. Es werden folgende Abkürzungen eingeführt: „bm“ für die Originaldaten aus [1], „org“ für das Multisinus-Signal mit Schroederphasen, „hom“ für das homogenisierte Multisinus-Signal und „hsc“ für das homogenisierte und skalierte Multisinus-Signal. Bild 3 zeigt die Ergebnisse der Homogenisierung, welche sowohl an den Zeitreihen, als auch am Histogramm deutlich zu erkennen ist. Bild 4 zeigt die bessere Abdeckung des Scheduling-Raums und Bild 5 die Häufigkeitsverteilungen der Ausgangssignale.

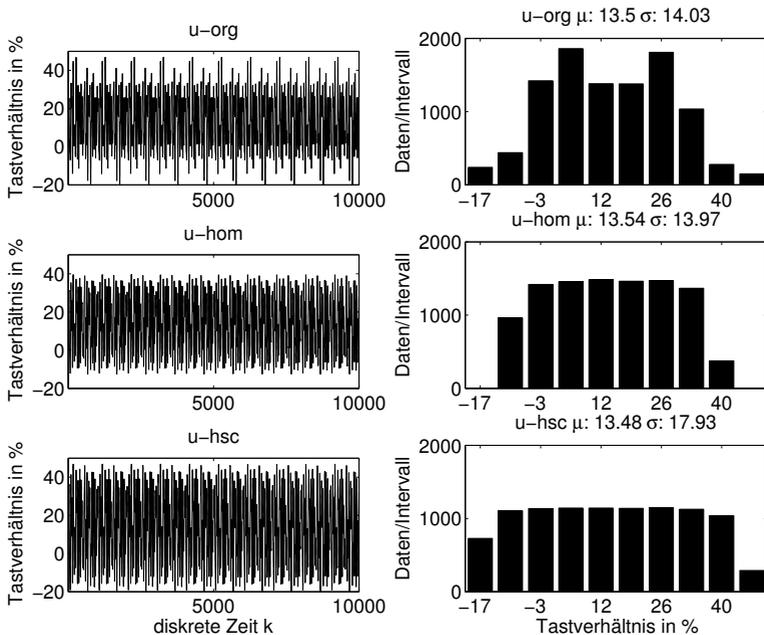


Bild 3: Gegenüberstellung verschiedener Testsignalvarianten

Beurteilung des Ausgangssignals

Da das nichtlineare Verhalten eines Modells, welches aus der Überlagerung linear affiner Teilmodelle besteht, allein durch die Partitionierung eingestellt wird, die Partitionierung in Form der Zugehörigkeitsfunktionen in den Modellansatz eingeht und die Zugehörigkeitsfunktionen von der Scheduling-Variable abhängen, spielt zum einen die Wahl der Scheduling-Variablen selbst und zum anderen die Abbildung des relevanten dynamischen Verhaltens im Scheduling-Raum eine große Rolle. Es war zu untersuchen, ob durch die Anwendung der Homogenisierungsmethode eine gleichmäßigere Verteilung im Scheduling-Raum zustandekommt und ob ein Simulationsmodell höherer Qualität entsteht, also ob die gleichmäßigere Verteilung des Eingangssignals mit einer verbesserten Abbildung des dynamischen Verhaltens korrespondiert. Zur Bewertung der Identifikation wurde das Validierungssignal aus [3] genutzt. Bild 4 zeigt die Identifikationsdaten der vier Fälle im Scheduling-Raum inklusive der mit dem FCM-Algorithmus bestimmten Prototypen sowie den Prototypen, welche sich aus der nichtlinearen Optimierung ergeben haben. Es sei angemerkt, dass die Anzahl der dargestellten Daten für jeden Fall gleich groß ist. Die aus [3] übernommenen Messdaten (Bild 4 links oben) zeigen, dass sich die Daten in der Mitte häufen, was auch mit dem in Bild 3 dargestellten originalen Multisinus-Signal übereinstimmt. Dort werden die Spitzenwerte nur vereinzelt erreicht. Daher ist die Abdeckung in dem Bereich auch nicht sehr dicht. Unter Berücksichtigung der anderen im Bild 4 enthaltenen Plots lässt sich außerdem feststellen, dass sich die Lage der Prototypen durch die Homogenisierung nur unwesentlich ändert. Bei der Benutzung desselben Testsignals bei dem anderen Prüfling (Bild 4 rechts oben) ist eine deutliche Ähnlichkeit zu den Originaldaten festzustellen. Die Unterschiede sowie das Ausreißen zweier der Prototypen durch die Optimierung sind auf den ersetzten Prüfling zurückzuführen. Die Abdeckung des Scheduling-Raums hat sich durch die Anregung der Drosselklappe mit einem homogenisierten Multisinus-Signal (Bild 4 links unten) deutlich verbessert. Es ist sowohl bei diesem als auch beim Fall des nachskalierten Signals (Bild 4 rechts unten) zu erkennen, dass die Lage der Prototypen sich drastisch ändert. Durch die Homogenisierung des Eingangssignals soll sich eine gleichmäßigere Anregung des Systems einstellen, was in der Scheduling-Variable deutlich werden muss. Werden aber die Plots aus Bild 5 betrachtet, welche die Verteilung der Ausgangsgröße zeigen, wird deutlich, dass eine gleichmäßigere Verteilung des Eingangssignals keine gleichmäßigere Verteilung der Ausgangsgröße zur Folge hat, was entsprechend der getroffenen Annahme durch das Tiefpassverhalten erklärt werden kann. Zudem folgt die Verbesserung der Abdeckung des Scheduling-Raums daraus, dass das homogenisierte Eingangssignal eine Komponente der Scheduling-Variable ist.

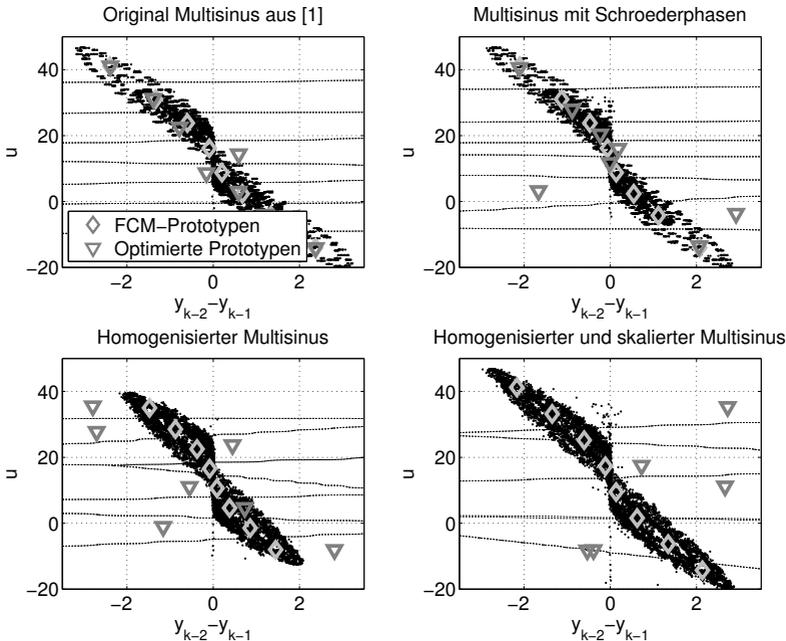


Bild 4: Gegenüberstellung Abdeckung des Scheduling-Raums

Identifikationsergebnisse

Im letzten Schritt wird die Güte der identifizierten Modelle anhand der eingeführten Kriterien beurteilt. Dazu werden die Maße sowohl bezüglich des Validierungs- und Identifikationsdatensatzes (Bilder 6 und 7) als auch der Art der Parameterbestimmung (Einschrittprädiktion oder auf Rekursion optimiert) ermittelt. Das Validierungssignal stammt aus [3] und es wurde wegen des ausgewechselten Prüflings ebenfalls ein neuer Versuch durchgeführt. An den Residuen der Identifikationsdaten ist zu erkennen, dass durch die Optimierung des Gesamtparametervektors in der rekursiven Modellauswertung (Bild 6 unten links und rechts) erwartungsgemäß ein deutlich besseres Modell entsteht. An den Residuen der Validierungsdaten ist zu erkennen, dass die Optimierung nicht den gewünschten Effekt hat. Zur Erklärung wird folgende Vermutung angestellt: Da das System Tiefpassverhalten zeigt und die Übertragung hoher Frequenzanteile abgeschwächt wird, sind in den gemessenen Systemantworten möglicherweise

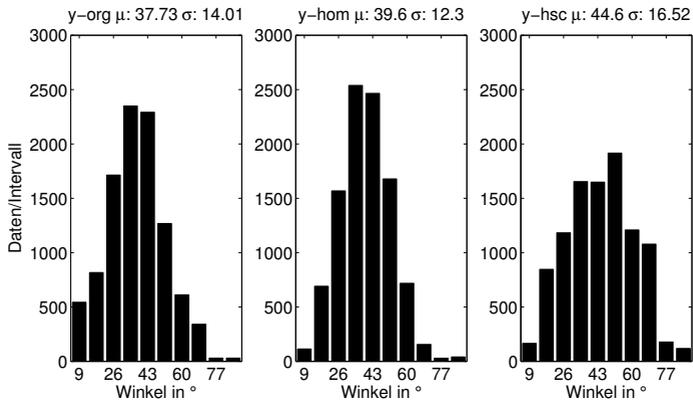


Bild 5: Verteilung der Ausgangssignale

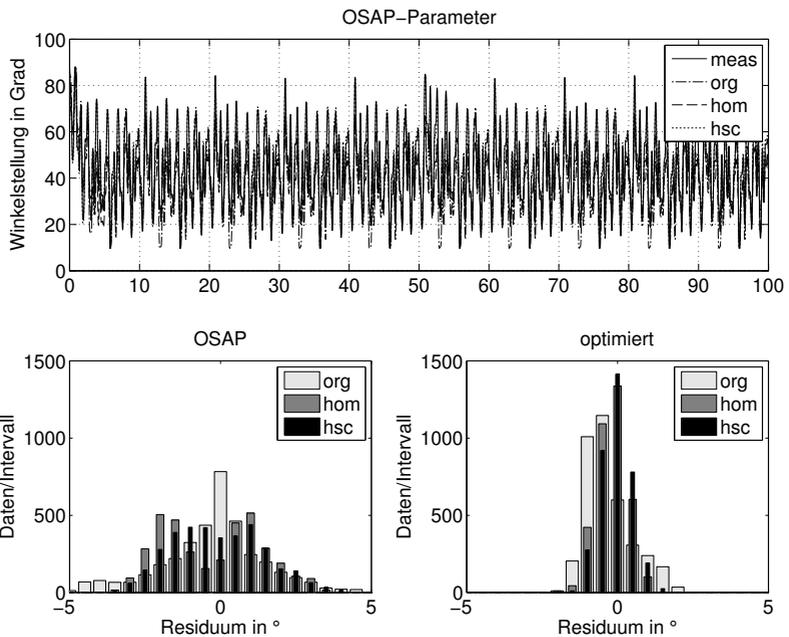


Bild 6: Identifikationsdaten mit Simulationsergebnissen

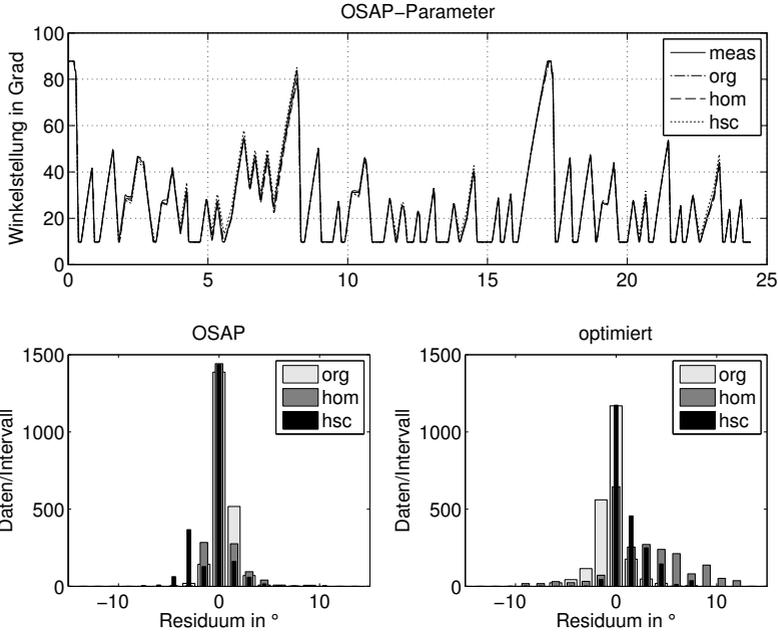


Bild 7: Validierungsdaten mit Simulationsergebnissen

relevante Dynamikanteile schwach vertreten. Die niederfrequenten Anteile nehmen deutlich höhere Werte an, weshalb sie bei der Schätzung der Parameter stärker ins Gewicht fallen. Dieser Effekt wird durch die Optimierung verstärkt. So werden Parameter, welche mit der höherfrequenten Dynamik korrespondieren, schlechter. Bei einer Systemanregung, in welcher die niederfrequenten Anteile nicht mit deutlich höheren Amplituden auftreten, wird die geringere Genauigkeit der mit der höherfrequenten Dynamik korrespondierenden Parameter deutlich. Tabelle 1 zeigt die Gütewerte der identifizierten Modelle. Auch an den Maßen ist zu erkennen, dass die Optimierung nicht den gewünschten Effekt hat, die Homogenisierungsmethode (ohne Skalierung) jedoch bei der Simulation mit Einzschrittprädiktionsparametern in der Regel eine Verbesserung bringt. Dieser Effekt kann bei der Homogenisierungsmethode mit Skalierung nicht beobachtet werden. Mit der Identifikation konnten außerdem die Ergebnisse aus [3] reproduziert werden.

Da der Scheduling-Raum mit homogenisiertem und skaliertem Eingangssignal (hsc) auch bei einer Anzahl von $c = 8$ Teilmodellen nur in sechs wirksame Partionionen unterteilt wird, wurde eine Identifikation mit $c = 6$ Teilmodellen wiederholt. Die Gütemaße dieser Identifikation ($\text{hsc}_{c=6}$) finden auch sich in der Tabelle. Es konnte zwar im Vergleich zu acht Teilmodellen eine bessere Simulationsfähigkeit auf den Identifikationsdaten bezüglich der optimierten Parameter erreicht werden (ca. 2 % (RMSE), ca. 7 % (MAE)), jedoch fallen alle weiteren Gütekriterien deutlich schlechter aus (Abweichungen von ca. 6 % bis 99 %).

Tabelle 1: Gütwerte der identifizierten Modelle in $^\circ$

	Identifikationsdaten				Validierungsdaten			
	OSAP		optimiert		OSAP		optimiert	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
$\text{bm}_{c=8}$	1,4	4,43	1,04	4,0	1,06	3,92	1,74	8,17
$\text{org}_{c=8}$	1,77	6,01	0,67	2,19	1,23	9,53	1,64	7,13
$\text{hom}_{c=8}$	1,23	3,65	0,45	1,95	1,27	7,06	4,87	12,68
$\text{hsc}_{c=8}$	1,75	6,43	0,45	1,66	1,89	10,95	2,1	10,03
$\text{hsc}_{c=6}$	1,81	6,83	0,44	1,54	5,35	16,73	6,5	20,14

Zusammenfassung und Ausblick

Dieser Beitrag hat das Problem des Erzeugens informationsreicher Daten für die Systemidentifikation ohne Vorwissen über das Modell anzunehmen angesprochen. Dazu wurde die Homogenisierungsmethode für die nichtlineare Systemidentifikation untersucht: Sie ändert Multisinus-Signale als Vertreter der Breitbandtestsignale für lineare dynamische Systeme so, dass sie gleichverteilt sind. Die Wirkung der Methode wurde anhand eines einfachen linearen Systems diskutiert, bevor die Identifikation von TS-Modellen am Beispiel eines elektromechanischen Stellantriebs durchgeführt wurde. Die Homogenisierungsmethode führt stellenweise zu Verbesserungen. Die ersten punktuellen Ergebnisse der Fallstudie sollen im nächsten Schritt erweitert und eine weitere technische Anwendung analog untersucht werden.

Literatur

- [1] Z. Ren, „Zur Identifikation mechatronischer Stellglieder mit Reibung bei Kraftfahrzeugen,“ Dissertation, FG Mess- und Regelungstechnik, Universität Kassel, 2012.
- [2] A. Kroll, H. Schulte, „Benchmark problems for nonlinear system identification and control using Soft Computing methods: Need and overview,“ *Applied Soft Computing*, Band 25, S. 496–513, 2014.
- [3] S. Zaidi, A. Kroll, „Electro-mechanical throttle as a benchmark problem for nonlinear system identification with friction,“ in *Proc. 24. Workshop Computational Intelligence*, Dortmund, S. 173–186, KIT Scientific Publishing, 2014.
- [4] Z. Ren, A. Kroll, F. Sofsky, F. Laubenstein, „On identification of piecewise-affine models for systems with friction and its application to electro-mechanical throttles,“ in *Proc. 16th IFAC Symposium on System Identification*, Brüssel, S. 1395–1400, 2012.
- [5] S. Duym, J. Schoukens, „Design of excitation signals for the restoring force surface method,“ *Mechanical Systems and Signal Processing*, Band 9, Nr. 2, S. 139–158, 1995.
- [6] R. Pintelon, J. Schoukens, *System identification: A frequency domain approach*, 2. Auflage. New York: Wiley-IEEE Press, 2012.
- [7] C. Hametner, M. Stadlbauer, M. Deregnaucourt, M. Jakubek, T. Winsel, „Optimal experiment design based on local model networks and multilayer perceptron networks,“ *Engineering Applications of Artificial Intelligence*, Band 26, Nr. 1, S. 251–261, 2013.
- [8] B. Hartmann, T. Ebert, O. Nelles, „Model-based design of experiments based on local model networks for nonlinear processes with low noise levels,“ in *Proc. American Control Conference*, San Francisco, CA, S. 5306–5311, 2011.
- [9] K. S. Narendra, „Identification and control of dynamical systems using neural networks,“ *IEEE Transactions on Neural Networks*, Band 1, Nr. 1, S. 4–27, 1990.

Vorwissen in Funktionsapproximationen durch Support-Vektor-Regression bei schlechter Datenqualität

Wolfgang Doneit, Ralf Mikut, Lutz Gröll, Markus Reischl

Karlsruher Institut für Technologie, Institut für Angewandte Informatik
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen
E-Mail: wolfgang.doneit@kit.edu

Motivation

Häufig stammen Datensätze für eine datengetriebene Modellierung aus Messungen oder Simulationen. Das Generieren der Datentupel kann für manche Wertebereiche der Rohmerkmale sehr aufwändig und kostenintensiv sein. Eine Reduktion von Aufwand und Kosten der Datengewinnung führt häufig zu einer ungleichmäßigen Verteilung der Daten [1] und/oder einer schlechten Güte komplexer Modelle [2]. Zudem kann sich der Einfluss einzelner Datentupel erhöhen [3]. Weitere Aspekte schlechter Datenqualität sind inkonsistente Datenquellen und Messfehler in Form von Ausreißern oder sich widersprechender Messungen. Betrachtet werden datengetriebene Funktionsapproximationen, bei denen der wahre Zusammenhang zwischen den Eingangsgrößen und der Ausgangsgröße unbekannt ist und die Ausgangsgröße einen signifikanten Störterm beinhalten kann. Um solche Effekte zu kompensieren, kann Vorwissen in die Modellbildung integriert werden. Einen Überblick über die Formulierungszugänge von Vorwissen in der Modellbildung gibt [4].

In manchen Anwendungen ist das Vorwissen als Funktionsklasse gegeben, wodurch der generelle Verlauf der approximierenden Funktion feststeht. Es werden jedoch häufig Polynomansätze oder Black-Box-Methoden wie Künstliche Neuronale Netze verwendet, die sich unabhängig von einem für den Anwendungszweck geeigneten Funktionsverlauf an die vorliegenden Daten anpassen. Es liegt eine ausschließlich datengetriebene Modellbildung vor. Auf Seiten der Datengewinnung (Domänenexperte der zu untersuchenden Problemstellung) können Informationen – das Vorwissen – vorhanden sein, welche in die Funktionsapproximation integriert werden können. Dieses Vorwissen bleibt in der Modellbildung oft ungenutzt, weil es nicht vom Domänenexperten erfragt wird. Das mag daran liegen, dass beispielsweise die Nichtnegativität der Modellfunktion noch intuitiv formulierbar ist, die

Erfassung von zulässigen Parameterbereichen, maximalen partiellen Ableitungen u.ä. wird allerdings aufgrund der Komplexität häufig vernachlässigt und führt zum Verlust von de facto vorhandenen Informationen.

Dieser Beitrag zeigt, wie Vorwissen über den Funktionsverlauf praxisnah und systematisch

- beim Domänenexperten erfasst,
- in Gleichungen übersetzt und
- in die Modellbildung integriert werden kann.

Als Beispiel zur Modellbildung dient die Support-Vektor-Regression mit Gauß-Kernen.

Sämtliche Berechnungen werden in MATLAB und Gait-CAD [5] durchgeführt.

Stand der Technik

Funktionsapproximation

Gegeben sei eine aus N Datentupeln bestehende Datenmatrix $\mathbf{X} \in \mathbb{R}^{N \times d}$, welche die Werte der Eingangsgrößen beinhaltet, sowie ein Ausgangsgrößenvektor $\mathbf{y} \in \mathbb{R}^{N \times 1}$, welcher die zugehörigen fehlerbehafteten Werte der Ausgangsgröße enthält. Gesucht ist ein Modell, das sowohl die vorliegenden Daten approximiert, als auch eine Generalisierungsfähigkeit aufweist, die Prädiktionen für bislang nicht in \mathbf{X} erfasste Werte der Eingangsgrößen ermöglicht.

Anders als in der Regressionsanalyse der Statistik, ist die den Daten zugrundeliegende Funktionsstruktur nicht bekannt. Die Problemstellung wird in diesem Beitrag *Funktionsapproximation* genannt⁴. Es handelt sich dabei meist um einen iterativen Prozess aus Strukturwahl und Parameteranpassung.

Hierfür kann auf eine breite Vielfalt an Methoden und Ansätzen zurückgegriffen werden, die aus Sicht der Datenqualität zunächst je einen besonderen Aspekt berücksichtigen. Ihre Auswahl erfolgt daher mit Hilfe von Vorwissen.

Bei Störtermen in den Eingangsgrößen finden *Error-In-Variables*-Modelle und *Total-Least-Squares*-Ansätze [6] Verwendung. In der *Robust Regression* mindern spezielle loss-Funktionen den Einfluss von Ausreißern auf die Modellbildung [7].

⁴ In Anlehnung an [4] kann ein solches Problem *Approssession* genannt werden. Der Begriff verdeutlicht die Kombination aus Approximation und Regression.

Tikhonov-Regularisierungen dienen in der *Ridge Regression* der Kompensation negativer Effekte von Multikollinearitäten zwischen Regressoren. Abseits klassischer statistischer Verfahren passen *Local Linear Model Trees* (LOLIMOT) bei schwankender Komplexität über dem Eingangsraum ihre Modellstruktur lokal den Daten an [8]. *Takagi-Sugeno-Kang Fuzzy Systems* stellen lokale Modelle dar, deren Gültigkeit im Eingangsraum durch unscharfe Regeln bestimmt wird [9].

Die Verfahren können kombiniert werden, um mehrere Aspekte gleichzeitig zu berücksichtigen. In diesem Beitrag wird die Support-Vektor-Regression (SVR) verwendet, auf deren Funktionsweise und Vorteile in einem späteren Abschnitt eingegangen wird.

Vorwissen über Eigenschaften und den Verlauf der gesuchten Funktion kann mit Hilfe verschiedener Formulierungszugänge in die Modellbildung integriert werden [4]:

- expliziter Zugang über Restriktionen,
- impliziter Zugang über Strukturansätze und
- Kompromisszugang über Strafterme.

In [4] dient als Beispiel das Vorwissen über die Nichtnegativität einer Funktion $f: \mathcal{D} \rightarrow \mathbb{R}$. Der *explizite Zugang über Restriktionen* $\forall x \in \mathcal{D} : f(x; \boldsymbol{\theta}) \geq 0$ oder relaxiert durch eine endliche Anzahl an Restriktionen $f(x_k) \geq 0; k = 1, \dots, N$ bietet den Vorteil, dass die zu optimierende Funktion der Parameteranpassung unverändert bleibt. Es muss jedoch ein restringiertes Optimierungsproblem gelöst werden.

Beim *impliziten Zugang über einen Strukturansatz* $f(x; \boldsymbol{\theta}) = [g(x; \boldsymbol{\theta})]^2$, werden zwar keine Restriktionen benötigt, um die Nichtnegativität zu sichern, es kommt jedoch häufig zum Konvexitätsverlust des Optimierungsproblems. So gestaltet sich z.B. für $g(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x$ das Least-Squares-Problem als Minimierung der Gütefunktion $Q(\boldsymbol{\theta}) = \sum_{i=1}^N (y_i - (\theta_0 + \theta_1 x_i)^2)^2$.

Als *Kompromisszugang* bietet die Gütefunktion

$$Q(\boldsymbol{\theta}) = \sum_{i=1}^N (y_i - f(x_i; \boldsymbol{\theta}))^2 + \sum_{k=1}^M \gamma_k (\min\{0, f(x_k; \boldsymbol{\theta})\})^2 \quad (1)$$

die Möglichkeit, das Vorwissen ohne eine zwingende Einhaltung einzubeziehen. Den Grad der Einhaltung bestimmen hier die Wichtungsfaktoren γ_k . Wie am Beispiel der Nichtnegativität gesehen, können sich Eigenschaften auf Bereiche des Eingangsraums beziehen, lassen sich aber bei geeigneter Wahl der Stützstellen

auch durch eine endliche Anzahl punktwiser Anforderungen sicherstellen. Die Anzahl, respektive die Dichte der Stützstellen, muss abhängig von der Funktionsstruktur ausreichend groß gewählt werden. Es bietet sich außerdem an, die Stützstellen im betrachteten Bereich beispielsweise äquidistant zu verteilen.

Das resultierende Optimierungsproblem muss anschließend gelöst, bzw. zuvor vereinfacht werden. Dazu können beispielsweise Reduktions- oder Erweiterungsmethoden verwendet werden.

Support-Vektor-Regression

Support-Vektor-Maschinen wurden ursprünglich für Klassifikationsaufgaben entwickelt [10, 11]. Ihr Anwendungsfeld wurde allerdings auch auf Regressionsprobleme erweitert [12]. Bei der SVR wird eine Funktion $f : \mathcal{X} \rightarrow \mathbb{R}$ gesucht, die für jeden Eingangsgrößenvektor \mathbf{x}_i (i -te Zeile der Datenmatrix \mathbf{X}) höchstens um den Wert ϵ vom zugehörigen Wert y_i des Ausgangsgrößenvektors abweicht und dabei den Parametervektor $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_N]$ minimiert. Mit einer ϵ -insensitiven loss-Funktion werden Abweichungen ermöglicht, die größer als ϵ sind. Nicht-lineare Zusammenhänge lassen sich mit Hilfe von Kerneltransformationen und einer entsprechenden Kernelfunktion $k(.,.) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ abbilden. Bei der Verwendung eines Gauß-Kernels lautet die Funktionsvorschrift der Kernelfunktion $k(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\boldsymbol{\Sigma}_K})$. Für die gesuchte Funktion ergibt sich die Struktur:

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b. \quad (2)$$

Die Wichtung zwischen den beiden Zielen

- Approximation der Daten und
- Minimierung der Norm des Parametervektors

wird durch die Wahl des Regularisierungsparameters C vorgenommen. Im Optimierungsproblem bestimmt er den maximal zulässigen Betrag der Parameter $\alpha_i \in [-\frac{C}{N}, \frac{C}{N}]$, $\forall i = 1, \dots, N$. Für eine ausführliche Herleitung des resultierenden Optimierungsproblems in der SVR wird auf [13] verwiesen. In der herkömmlichen SVR mit Gauß-Kernel müssen vor der Parameteranpassung folgende Metaparameter gewählt werden:

- Varianz σ_K bzw. Kovarianzmatrix $\boldsymbol{\Sigma}_K$ des Gauß-Kernels,
- Breite ϵ des insensitiven Bereichs der loss-Funktion und

- Regularisierungsparameter C .

Die eigentliche Parameteranpassung von $\alpha = [\alpha_1, \dots, \alpha_N]$ stellt ein konvexes Optimierungsproblem dar, was die Verwendung herkömmlicher Lösungsalgorithmen ermöglicht.

In [14, 15, 16] wird unter anderem gezeigt, wie Vorwissen über Funktionswerte und Ableitungen in die Support-Vektor-Regression integriert werden können.

Dieser Beitrag zeigt eine Vorgehensweise, wie Vorwissen über die Struktur, den Verlauf und über Eigenschaften einer gesuchten, approximierenden Funktion beim Domänenexperten erfasst wird, um es als mathematische Forderungen formulieren und am Beispiel der Support-Vektor-Regression in die Modellbildung integrieren zu können. Eine strukturierte Überführung der verbalen Beschreibung einer Funktion bis in das Optimierungsproblem der Funktionsapproximation dient einer Verbesserung der Generalisierungsfähigkeit durch ein größtmögliches Ausnutzen des vorhandenen Vorwissens.

Methoden

Allgemeines

Bild 1 zeigt eine schematische Darstellung der datengetriebenen und expertengestützten Funktionsapproximation. Die Besonderheit des Schemas ist die detaillierte Betrachtung des Einflusses von Vorwissen in die Modellbildung. Eine zentrale Rolle nimmt dabei die Erfassung von Vorwissen ein. Die hier gewonnenen Informationen fließen auf drei Arten in die Modellbildung ein: In die Bewertung der Datenqualität sowie der daraus resultierenden Datentupel- und Merkmalsselektion und Datenkorrektur, in die Formulierung des Optimierungsproblems und in die Bewertung der Modellqualität.

Z.B. kann für eine Datentupelselektion die Wahl eines geeigneten Ausreißererkennungsvorgahens häufig nur mit Expertenwissen über Verteilungsannahmen etc. getroffen werden. Die Zuverlässigkeit des Vorwissens wird bewertet und in der Modellbildung berücksichtigt, weil es statt als gesichertes Wissen auch als Vermutung oder Intuition vorliegen kann. Im Falle der Datentupelselektion kann dies beispielsweise statt dem Ausschluss aus dem Datensatz eine Wichtung einzelner Datentupel zur Folge haben.

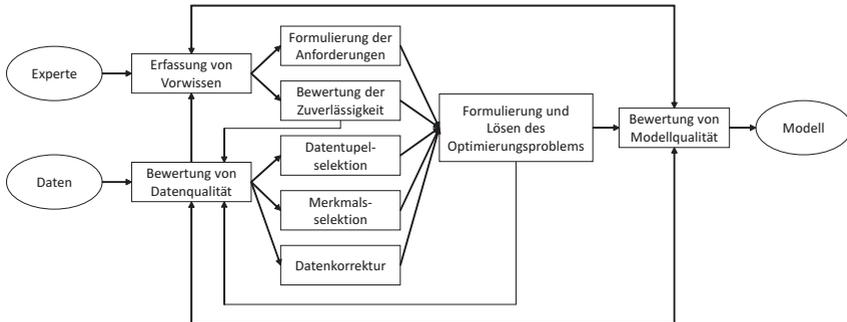


Bild 1: Abstrahiertes Schema des Modellbildungsprozesses

Nach der Formulierung der Anforderungen an die approximierende Funktion liegt das Vorwissen beispielsweise als mathematische Gleichungen und Ungleichungen vor, deren Zuverlässigkeit ebenfalls bewertet wurde. Das Vorwissen wird nun entsprechend der Approximationsmethode und der Zuverlässigkeit in die Formulierung des Optimierungsproblems integriert. Das Vorwissen kann hierbei auch der Wahl einer Funktionsklasse dienen.

In der Bewertung der Modellqualität kann Vorwissen beispielsweise der Wahl der Validierungsmethode dienen, oder das Modell wird durch Sichtprüfung anhand nicht-quantifizierbarer Eigenschaften bewertet. Wurde Vorwissen in die Formulierung des Optimierungsproblems integriert, muss in der Bewertung der Modellqualität zudem sowohl die eigentliche Datenapproximation, als auch die Erfüllung der Anforderungen bewertet werden.

Für diesen Beitrag wurde ein Fragebogen [17] entworfen⁵, um Vorwissen für Funktionsapproximationen zu erfassen. Die Fragen sind stellenweise redundant und mit beispielhaften Abbildungen verknüpft, um einen verständlichen Zugang zu bieten und möglichst viele Informationen vom Domänenexperten zu gewinnen. Die Erfassung bezieht sich auf die Eigenschaften: Kenntnis der Funktionsstruktur, Symmetrie, Grenzwertverhalten, Monotonie, Umkehr der Monotonie, Stetigkeit und Differenzierbarkeit, Definitions- und Wertebereich, Positivität und Negativität, Unimodalität und Konvexität, Glattheit, Integralrestriktionen sowie der Datenqualität (Ausreißer, Multikollinearitäten, heterogene Verteilungen).

Im folgenden Abschnitt wird ein Auszug des Fragebogens vorgestellt und das Vorwissen als Anforderungen formuliert. Die Integration des Vorwissens in die Formulierung des Optimierungsproblems wird am Beispiel der SVR gezeigt.

⁵ Download auf www.researchgate.net/profile/wolfgang_doneit/publications

Erfassung von Vorwissen und Formulierung der Anforderungen

Bild 2 zeigt Fragen über das Grenzwertverhalten von Funktionen.

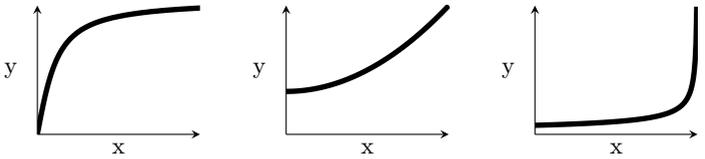
Verhalten für $x \rightarrow \infty$	3
<input type="checkbox"/> Liegt ein monotoner Verlauf mit gesättigtem Anstieg vor (a)?	
<input type="checkbox"/> Strebt die zweite Ableitung gegen null?	
<input type="checkbox"/> Flacht die Kurve ab (a)?	
<input type="checkbox"/> Erhöht sich die Steigung bis zu einem konstanten Wert (b)?	
<input type="checkbox"/> Wird die Steigung unendlich groß (c)?	
<input type="checkbox"/> Liegt eine Polstelle mit/ohne Vorzeichenwechsel vor (c)?	
<p style="text-align: center;">(a) (b) (c)</p> 	

Bild 2: Auszug des Fragebogens zur Gewinnung von Vorwissen über das Grenzwertverhalten der zu approximierenden Funktion.

Das Vorwissen, das aus der ersten Frage zum Grenzwertverhalten („Liegt ein monotoner Verlauf mit gesättigtem Anstieg vor?“) einer Funktion $f: \mathbb{R}^d \rightarrow \mathbb{R}, \mathbf{x} \rightarrow y$ generiert werden kann, besteht beispielsweise aus den bereichsweisen Anforderungen:

- f ist für $0 \leq x_k \leq 2$ in Richtung der Eingangsgröße x_k monoton steigend und
- der Anstieg von f ist für $1 \leq x_k \leq 2$ in Richtung der Eingangsgröße x_k gesättigt.

Die bereichsweisen Anforderungen lassen sich durch die unendlich vielen punktweisen Anforderungen im Definitionsbereich \mathcal{X} von f als

$$\forall \mathbf{x} \in \mathcal{X} | 0 \leq x_k \leq 2 : \frac{\partial f}{\partial x_k}(\mathbf{x}) \stackrel{!}{\geq} 0 \quad (3)$$

und

$$\forall \mathbf{x} \in \mathcal{X} | 1 \leq x_k \leq 2 : \frac{\partial f}{\partial x_k}(\mathbf{x}) \stackrel{!}{\geq} 0 \text{ sowie} \quad (4)$$

$$\forall \mathbf{x} \in \mathcal{X} | 1 \leq x_k \leq 2 : \frac{\partial f}{\partial x_k}(\mathbf{x}) \stackrel{!}{\leq} \tau; 0 \leq \tau \ll 1 \quad (5)$$

formulieren⁶. Für eine spätere Integration in das Optimierungsproblem werden die unendlich vielen punktweisen Anforderungen auf eine endliche Anzahl M_1 , bzw. $2M_2$ reduziert.

$$\frac{\partial f}{\partial x_k}(\mathbf{x}_j) \stackrel{!}{\geq} 0; j = 1, \dots, M_1 \quad (6)$$

$$\frac{\partial f}{\partial x_k}(\mathbf{x}_l) \stackrel{!}{\geq} 0; l = 1, \dots, M_2 \quad (7)$$

$$\frac{\partial f}{\partial x_k}(\mathbf{x}_l) \stackrel{!}{\leq} \tau; l = 1, \dots, M_2; 0 \leq \tau \ll 1 \quad (8)$$

Ein anderer Teil der Erfassung bezieht sich auf den Definitions- und Wertebereich der Funktion. Den entsprechenden Auszug des Fragebogens zeigt Bild 3. Der Definitionsbereich sei bereits als \mathcal{X} gegeben.

Das Vorwissen, das aus der zweiten Frage zum Definitions- und Wertebereich generiert werden kann, besteht beispielsweise aus der für \mathcal{X} globalen Anforderung:

- Die Funktionswerte von f liegen im Intervall $[0; 1]$.

Die globale Anforderung lässt sich durch die unendlich vielen punktweisen Anforderungen im Definitionsbereich \mathcal{X} von f als

$$\forall \mathbf{x} \in \mathcal{X} : f(\mathbf{x}) \stackrel{!}{\geq} 0 \quad (9)$$

$$\forall \mathbf{x} \in \mathcal{X} : f(\mathbf{x}) \stackrel{!}{\leq} 1 \quad (10)$$

⁶ Das einem Vergleichsoperator übergestellte Ausrufezeichen zeigt die Forderung der entsprechenden Gleichung/Ungleichung an. Es ist zu lesen als „soll sein“.

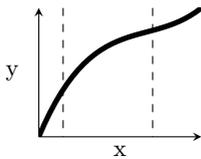
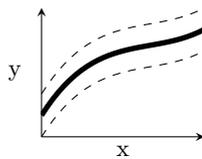
Definitionsbereich und Wertebereich	10
<input type="checkbox"/> Lässt sich der Definitionsbereich (die zulässigen, bzw. betrachteten Werte der Eingangsgrößen) einschränken (a)?	
<input type="checkbox"/> Lässt sich der Wertebereich (die zulässigen Funktionswerte) einschränken (b)?	
<input type="checkbox"/> Ist der zulässige maximale oder minimale Funktionswert von den Eingangsgrößen abhängig (b)?	
<input type="checkbox"/> Welcher Funktionswert darf global oder lokal nicht unter/überschritten werden?	
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>(a)</p>  </div> <div style="text-align: center;"> <p>(b)</p>  </div> </div>	

Bild 3: Auszug des Fragebogens zur Gewinnung von Vorwissen über den Definitions- und Wertebereich der zu approximierenden Funktion.

formulieren. Auch die globalen Anforderungen an den Wertebereich werden auf eine endliche Anzahl $2M_3$ punktweise Anforderungen reduziert.

$$f(\mathbf{x}_w) \stackrel{!}{\geq} 0; w = 1, \dots, M_3 \tag{11}$$

$$f(\mathbf{x}_w) \stackrel{!}{\leq} 1; w = 1, \dots, M_3 \tag{12}$$

Oftmals bietet es sich an, Regularisierungen zu verwenden, um einen gefälligen Verlauf der approximierenden Funktion zu erhalten.

Mathematisch entspricht das einer Reduktion oder Beschränkung der Krümmung auf den Wert τ , was beispielsweise mit

$$\forall \mathbf{x} \in \mathcal{X} : \left| \frac{\partial^2 f}{\partial x_k^2}(\mathbf{x}) \right| \stackrel{!}{\leq} \tau; \tau \geq 0 \tag{13}$$

formuliert werden kann und ebenfalls im Fragebogen erfasst wird. Die Forderung wird genutzt, um eine „Glattheit“ des Funktionsverlaufs sicherzustellen, bedingt aber nicht die unendlich oftmalige Differenzierbarkeit einer mathematisch glatten Funktion.

Zur Sicherstellung von Funktionseigenschaften kann es wichtig sein, explizit nicht die in der Datenmatrix \mathbf{X} vorhandenen Werte der Eingangsgrößen, sondern beliebige andere Werte als Stützstellen zu verwenden. Damit werden die Eigenschaften auch für Bereiche sichergestellt, die nicht in der Datenmatrix erfasst sind.

Vorwissen im Optimierungsproblem der Support-Vektor-Regression

Zur Integration von Vorwissen in die SVR werden im folgenden Abschnitt Anpassungen der Metaparameter ϵ und C vorgenommen. Bildet eine Funktionsapproximation f näherungsweise die wahre Funktionsstruktur ab, entsprechen die Abweichungen $y_i - f(\mathbf{x}_i)$ z.B. einer Störgröße in der Datengewinnung, die nicht durch f abgebildet werden soll. Andernfalls spricht man von Overfitting. In der Regressionsanalyse wird eine solche Störgröße häufig als normalverteilt mit Erwartungswert 0 angenommen, was eine symmetrische loss-Funktion begründet: Positive und negative Abweichungen werden gleich behandelt. Das gilt auch für ϵ -insensitive loss-Funktionen, wie sie in der SVR verwendet werden. Das für die SVR resultierende Optimierungsproblem lässt, beispielsweise in seiner Formulierung als quadratisches Programm, ohne Konvexitätsverlust eine für jedes Datentupel sowie für positive und negative Abweichungen von y_i individuelle Wahl von ϵ und C zu. Damit bestimmt C_i^+ den Strafterm für $f(\mathbf{x}_i) - y_i \geq \epsilon_i^+$ und C_i^- den Strafterm für $y_i - f(\mathbf{x}_i) \geq \epsilon_i^-$ mit $\epsilon_i^{+/-} \geq 0, i = 1, \dots, N$. Für eine punktweise Anforderung an den Funktionswert an einer Stützstelle \mathbf{x}_p können die Stützstellen und die entsprechenden geforderten Funktionswerte in die Datenmatrix und den Ausgangsgrößenvektor eingefügt werden:

$$\mathbf{X}^* = \begin{bmatrix} \mathbf{X} \\ \mathbf{X}_{\text{priori}} \end{bmatrix}, \mathbf{y}^* = \begin{bmatrix} \mathbf{y} \\ \mathbf{y}_{\text{priori}} \end{bmatrix} \quad (14)$$

Die zugehörigen Metaparameter können gemäß Tabelle 1 angepasst werden. Die Empfehlungen „groß“, bzw. „klein“ für $C_p^{+/-} / \epsilon_p^{+/-}$ beziehen sich auf die Mittelwerte der Parameter für die ursprünglichen Datentupel und müssen für jedes neue Datentupel gewählt werden. Mit der Wahl der Parameter wird die Zuverlässigkeit des Vorwissens modelliert. Alternativ kann ein einheitlicher Strafparameter C gewählt werden, wie es in der herkömmlichen SVR üblich ist. Eine besondere Wichtung der Anforderung gegenüber der gewöhnlichen Datenapproximation geht damit allerdings verloren. Entspricht N^* der Anzahl an Datentupeln nach

Integration der zusätzlichen Datentupel durch punktweise Anforderungen an Funktionswerte, ändern sich die zulässigen Grenzen der zu schätzenden Parameter:

$$\alpha_i \in \left[-\frac{C_i^-}{N^*}, \frac{C_i^+}{N^*} \right], \forall i = 1, \dots, N^* \quad (15)$$

Betrachtet wird nun wieder die Funktionsstruktur aus (2). Um punktweise Restriktionen bezüglich einer Ableitung von $f(\mathbf{x})$ zu beachten, kann die Funktion numerisch oder analytisch differenziert werden.

Die Ableitungen sind parameterlinear in α und nicht abhängig von b . Anforderungen an Ableitungen können demnach als Gleichheits- oder Ungleichheitsrestriktionen in die Randbedingungen des Optimierungsproblems eingebunden werden. Haben die Anforderungen an die Ableitung keine hohe Zuverlässigkeit, können weitere Schlupfvariablen eingeführt werden. Damit sind Abweichungen von der Anforderung an die Ableitung möglich, werden aber entsprechend ihrer Strafparameter in der Gütefunktion der Parameteranpassung bestraft.

Anforderung	C_p^+	C_p^-	ϵ_p^+	ϵ_p^-
$f(\mathbf{x}_p) \stackrel{!}{=} y_p$	groß	groß	klein	klein
$f(\mathbf{x}_p) \stackrel{!}{\leq} y_p$	groß	klein	klein	groß
$f(\mathbf{x}_p) \stackrel{!}{\geq} y_p$	klein	groß	groß	klein
$y_l \stackrel{!}{\leq} f(\mathbf{x}_p) \stackrel{!}{\leq} y_h$	groß	groß	$y_h - y_p$	$y_p - y_l$

Tabelle 1: Unterstützung zur Parameterwahl bei der Integration von punktweisen Anforderungen an Funktionswerte in SVR.

Ein Beispiel für schlechte Datenqualität kann die ungleichmäßige Verteilung der Daten im Eingangsraum sein, was zu Problemen in der Modellbildung führt. Es kommt entweder zu Underfitting in Bereichen vieler Daten und mit nicht-linearen Zusammenhängen oder zu Overfitting in Bereichen, in denen wenige Daten vorliegen. Ursache für lokales Overfitting in der SVR mit Gauß-Kerneln ist eine unvorteilhafte Parametrierung von C , ϵ und der Gauß-Kernel. In [2] wurde das Maß $Q_{IV, \text{total}}$ anhand der Datensätze D_1 und D_2 eingeführt, um Overfitting von Modellen in Bereichen zu bewerten, in denen wenige oder keine Daten vorliegen.

Die Bilder 4(a)-(c) zeigen Approximationen von Datensatz D_1 , die Bilder 4(d)-(f) zeigen Approximationen von Datensatz D_2 .

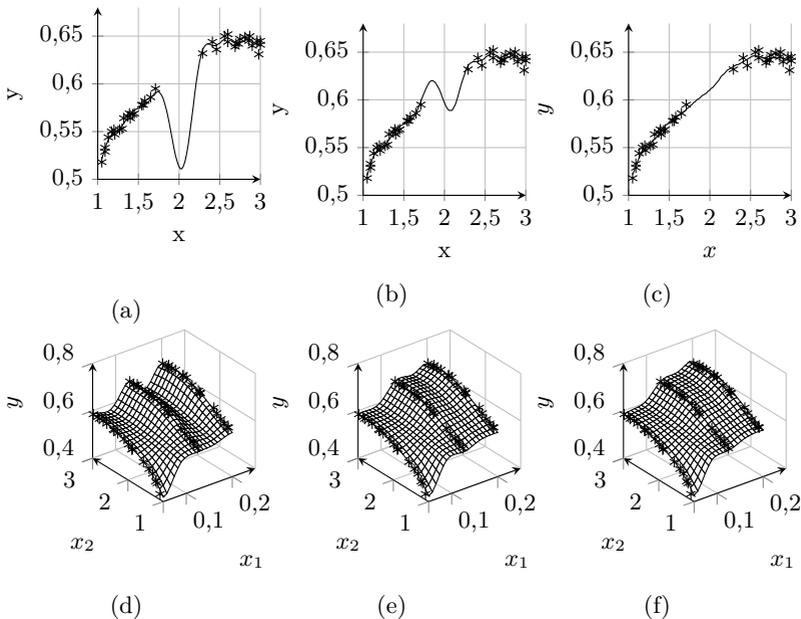


Bild 4: Schlecht parametrisierte Approximationen durch SVR mit Gauß-Kernel ohne ((a),(d)) und mit ((b),(c),(e),(f)) Integration von Vorwissen.

Bild 4(a) zeigt eine SVR mit Gauß-Kernel zu geringer Varianz. In Bereichen, in denen Daten vorliegen, tritt bereits Overfitting auf, für $x \in [1.7; 2.4]$ verliert die approximierende Funktion jegliche Generalisierungsfähigkeit und weist einen globalen Extremwert auf.

Für die Approximation aus Bild 4(b) wurden drei Datentupel im Intervall $[1.7; 2.4]$ des Definitionsbereichs imputiert, für die als Vorwissen lediglich bekannt ist, dass ihr Funktionswert im Intervall $[0.59; 0.63]$ liegen sollte (*Vorwissen 1*). Durch die Formulierung als Datentupel handelt es sich um keine Restriktion, deren Einhaltung erzwungen wird, sondern um einen Kompromisszugang mit Straftermen. Es ist zu beachten, dass die Imputation von Datentupeln die Anzahl der Datentupel erhöht. Da (15) von N^* abhängig ist, wird die Minimierung der Norm des Parametervektors α verstärkt.

Eine weitere Möglichkeit, die Generalisierungsfähigkeit des Modells zu steigern, ist eine Forderung von Monotonie für das Intervall $[1.7; 2.4]$ des Definitionsbereichs.

Sie wird punktweise als Restriktion eingeführt. Dadurch verschlechtert sich jedoch die Approximation an die Daten maßgeblich. Erst durch das zusätzliche Imputieren von 20 Datentupeln im betrachteten Intervall ergibt sich der Verlauf aus Bild 4(c) (*Vorwissen 2*).

Die Imputation einer so hohen Anzahl an Datentupeln gleicht allerdings einer Reparametrierung der SVR, bzw. der Strafparameter $C_i^{+/-}$, wodurch sich ein generell glatterer Verlauf der approximierenden Funktion ergibt, der in diesem Fall die Forderung von Monotonie bereits erfüllt.

Auch beim Modell in Bild 4(d) ist die Kovarianzmatrix der Gauß-Kernel schlecht gewählt, was am Overfitting zu erkennen ist. In Bild 4(e) werden mit Hilfe eines Nächste-Nachbarn-Verfahren lokale Wertebereiche definiert und 20 Datentupel imputiert, wodurch die Generalisierungsfähigkeit erhöht wird (*Vorwissen 3*). Für das Modell in Bild 4(f) wurde, zusätzlich zur Imputation der 20 Datentupeln, Monotonie in Richtung x_1 gefordert.

Tabelle 2 vergleicht die Modelle aus Bild 4 mit den Modellen durch Künstliche Neuronale Netze aus [2] für D_1 und D_2 anhand des *Root Mean Squared Error* über den zur Modellbildung zur Verfügung stehenden Daten (*in-sample-error*, IE) sowie durch das Bewertungsmaß $Q_{IV,total}$ zur Bewertung des Modells für $x \in [1.7; 2.4]$, bzw. für die Umgebungen von $\mathbf{x} = (0.09 \ 2)$ und $\mathbf{x} = (0.17 \ 2.4)$. Der Vorteil der Integration von Vorwissen zeigt sich durch eine deutliche Verbesserung von $Q_{IV,total}$ bei meist nur geringer Verschlechterung des *IE* gegenüber den Modellen ohne Vorwissen. Nur beim Modell mit *Vorwissen 4* kommt es zu einer deutlichen Verschlechterung der Approximationsgüte. Mit ausführlich formuliertem Vorwissen kann zwischen der Erfüllung der Monotonie und der Approximationsgüte entschieden werden. Das Overfitting ist zudem bei den Modellen durch SVR weniger ausgeprägt als bei den Modellen durch KNNs.

Anwendung

In realen Anwendungen ist das Vorwissen häufig komplexer und vielfältiger. Bei der Kalibrierung von Mensch-Maschine-Schnittstellen auf Basis von zweikanaligen Messungen von Muskelaktivitäten (EMG-Signale) werden Regressionsmodelle verwendet, um von gemessenen EMG-Signalen auf die vom Anwender intentionierten Aktivitätssignale der Kanäle zu schließen [18]. Die Funktionen $f_1(\mathbf{x}; \beta_1) : \mathbb{R}^2 \rightarrow \mathbb{R}, \mathbf{x} \rightarrow y_1$ und $f_2(\mathbf{x}; \beta_2) : \mathbb{R}^2 \rightarrow \mathbb{R}, \mathbf{x} \rightarrow y_2$ sollen dabei aufgezeichnete Kalibrierungsdaten approximieren. Die Datensätze bestehen aus diskretisierten zeitlichen Abtastungen normierter Aktivitätssignale $\mathbf{x} = (x_1, x_2)$, denen zwei Ausgangsgrößen y_1 und y_2 zugeordnet sind. Die Daten stammen aus

einer Kalibrierungsroutine: Die Werte der Ausgangsgrößen waren Soll-Vorgaben, die ein Anwender durch entsprechende Aktivierung von Muskeln während der folgenden Kalibrierungsschritte zu erfüllen versuchte:

- maximale (1) und halbe (2) Aktivierung Kanal 1,
- maximale (3) und halbe (4) Aktivierung Kanal 2 und
- maximale (5) und halbe (6) Ko-Aktivierung

Im Folgenden wird die Approximationsfunktion f_1 gesucht.

Datensatz	Approximation	IE	$Q_{IV,total,T2}$	
D_1	KNN 1 Neuron [2]	6.7	0.94	
D_1	KNN 3 Neuronen [2]	4.9	0.96	
D_1	KNN 5 Neuronen [2]	4.4	0.85	
D_1	SVR ohne Vorwissen	3.6	0.23	
D_1	SVR mit <i>Vorwissen 1</i>	3.6	0.45	
D_1	SVR mit <i>Vorwissen 2</i>	3.9	0.92	
Datensatz	Approximation	IE	$Q_{IV,total,T1}$	$Q_{IV,total,T4}$
D_2	KNN 3 Neuronen [2]	7.8	0.82	0.92
D_2	KNN 7 Neuronen [2]	6.1	0.3	0.3
D_2	KNN 14 Neuronen [2]	4.0	0.54	0.27
D_2	SVR ohne Vorwissen	4.0	0.79	0.25
D_2	SVR mit <i>Vorwissen 3</i>	4.2	0.85	0.7
D_2	SVR mit <i>Vorwissen 4</i>	7.9	0.86	0.81

Tabelle 2: Vergleich von Modellen anhand von IE und des Bewertungsmaßes $Q_{IV,total}$ zur Bewertung von Overfitting [2] für D_1 und D_2 .

Die sechs Kalibrierungsschritte für die Approximation von f_1 liefern eine Datenverteilung im zweidimensionalen Eingangsraum wie in Bild 5(a). Durch Vorwissen werden mit Hilfe von Normalverteilungsannahmen Ausreißer entfernt und die Datencluster der Kalibrierungsschritte anschließend ausgedünnt, um eine unausgeglichene Verteilung der Daten auf die einzelnen Cluster zu vermeiden. Man erhält einen Datensatz, wie er in Bild 5(b) zu sehen ist. Für die Anwendung unerlässlich ist der Funktionswert $f_1(0,0) = 0$, um bei keiner gemessenen Aktivität beider Kanäle auch keine partiellen Aktivitätssignale an Steuergeräte o.ä. zu senden. Dieser Fall wird nicht durch die Kalibrierungsschritte abgedeckt, weshalb die Daten durch Vorwissen künstlich imputiert werden müssen.

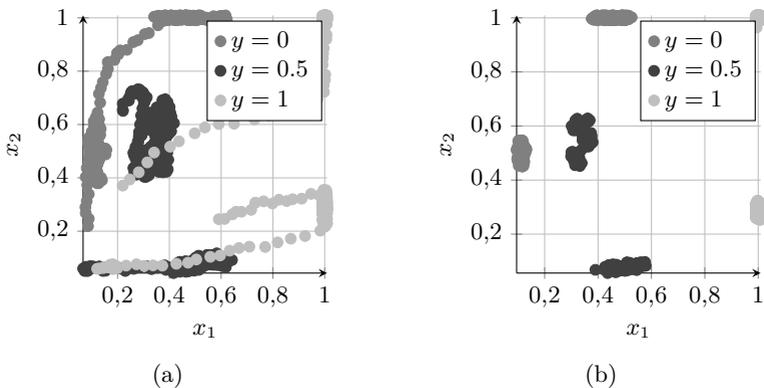


Bild 5: (a) zeigt den Eingangsraum mit den Rohdaten und (b) zeigt den mit Hilfe von Vorwissen ausreißerbereinigten und ausgedünnten Datensatz

Mit Hilfe des Fragebogens zur Erfassung von Vorwissen wird folgendes Vorwissen erfasst:

- Ist der Verlauf steigend?
Ja, in Richtung der Eingangsgröße x_1 .
- Darf der Funktionsverlauf „Sprünge“ enthalten?
Nein.
- Darf der Funktionsverlauf „Knicke“ enthalten?
Ja.
- Wird die approximierende Funktion noch weiterverarbeitet?
Ja, aus Sicherheitsgründen werden alle Funktionswerte < 0 auf 0 und alle Funktionswerte > 1 auf 1 gesetzt.
- Lässt sich der Definitionsbereich einschränken?
Ja, es werden nur Eingangsgrößen $x_{1/2} \in [0; 1]$ betrachtet.
- Lässt sich der Wertebereich einschränken?
Ja, wird allerdings durch Nachverarbeitung auf jeden Fall erreicht.

- Ist das Verhalten der Funktion für bestimmte Werte der Eingangsgrößen bekannt?

Ja, $f_1(0, 0) \stackrel{!}{=} 0$.

- Gibt es Bereiche in denen Negativität oder Nichtnegativität der gesuchten Funktion gefordert sind?

Ja, die Funktion muss im gesamten Definitionsbereich ≥ 0 sein.

- Ist die gesuchte Funktion (streng) monoton bezüglich eines Regressors?

Ja, monoton steigend bezüglich x_1 .

- Existieren Werte für partielle Ableitungen, die nicht über- oder unterschritten werden dürfen?

Ja, aber in Abhängigkeit von der Standardabweichung der Datencluster im Eingangsraum.

- Ist der gewünschte Funktionsverlauf glatt?

Ja, im Sinne einer geringen Krümmung.

Mit den vorgestellten Methoden wurde das Vorwissen in die Modellbildung integriert.

Kalibrierungsschritt	MAE ohne Vorwissen	MAE mit Vorwissen
1	0	0
2	0.0005	0.0144
3	0.0007	0
4	0.0115	0
5	0	0
6	0.0188	0.0721

Tabelle 3: Mittlere absolute Fehler (MAE) für die Datencluster bei einem Modell ohne bzw. mit Integration von Vorwissen.

Neben der Monotonierestriktion wurden vor allem asymmetrische insensitive Bereiche in der loss-Funktion für Datentupel mit $y = 0$ bzw. $y = 1$ verwendet. Dazu wird das Wissen über die Nachverarbeitung und der nicht geforderten Differenzierbarkeit verwendet. Tabelle 3 stellt die mittleren absoluten Fehler (MAE) eines Modells ohne Integration von Vorwissen und eines Modells mit

Integration von Vorwissen für die sechs Datencluster gegenüber. Die getrennte Auflistung der Fehler zeigt, in welchen Bereichen sich die Näherung an die Daten durch das Vorwissen verbessert bzw. verschlechtert hat.

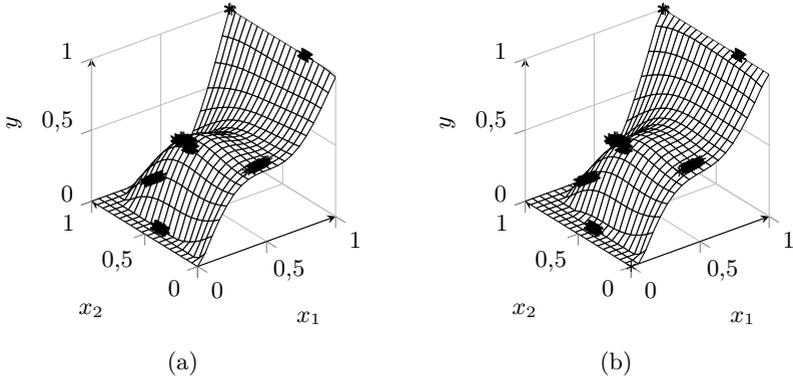


Bild 6: (a) zeigt ein Modell ohne Integration von Vorwissen und (b) zeigt ein Modell mit Integration von Vorwissen

Bild 6(a) zeigt die entsprechende approximierende Funktion ohne Integration von Vorwissen und Bild 6(b) die approximierende Funktion mit Integration von Vorwissen. Das Modell, in dem Vorwissen integriert wurde, ist trotz der bereichsweisen schlechteren Näherung an die Daten für den Anwendungsfall vorzuziehen, da der partielle monotone Verlauf (z.B. im Bereich um $\mathbf{x} = (0.7 \ 0.5)$) für die Anwendung in einer Mensch-Maschine-Schnittstelle geeigneter ist. Das bedeutet, dass auch die Bewertung der Modellqualität nicht nur datengetrieben, sondern auch expertengestützt ist.

Zusammenfassung und Ausblick

In der datengetriebenen Approximation von Funktionen spielt vor allem für die Sicherstellung der Generalisierungsfähigkeit bei der Modellbildung die Integration von Vorwissen eine wichtige Rolle.

Integration von Vorwissen beschreibt dabei nicht nur das Einführen von Restriktionen in Optimierungsprobleme, sondern bezieht sich auf die Bewertung der

Datenqualität, die Formulierung des Optimierungsproblems und die Bewertung der Modellqualität.

Es wurden Auszüge aus einem Fragebogen vorgestellt, der bei Domänenexperten Vorwissen systematisch erfasst. Beispiele haben gezeigt, wie unterschiedliches Vorwissen über die tatsächliche Struktur der zu approximierenden Funktion bzw. ihren Verlauf zunächst als mathematische Gleichungen und Ungleichungen formuliert und anschließend in eine Support-Vektor-Regression integriert werden kann. Der Nutzen von unsicherem Vorwissen über lokale Wertebereiche wurde anhand von zwei simulierten Datensätzen gezeigt. Mit einem Anwendungsbeispiel wurde die Notwendigkeit der systematischen Erfassung von Vorwissen aufgezeigt.

Nächste Schritte sind die Verwendung des Fragebogens und des daraus entstehenden Vorwissens in weiteren realen Anwendungsfällen. Außerdem soll geprüft werden, wie sich bereichsweise Anforderungen (z.B. Monotonie) auf Benchmarkprobleme gemäß [19] auswirken.

Literatur

- [1] Forrester, A.; Sobester, A.; Keane, A.: *Engineering Design via Surrogate Modelling: a Practical Guide*. John Wiley & Sons. 2008.
- [2] Doneit, W.; Mikut, R.; Pychynski, T.; Reischl, M.: Abstands- und Monotonie Maße für Regressionsmodelle mit heterogenen Lerndaten. In: *Proc., 24. Workshop Computational Intelligence, Dortmund*, S. 1–16. KIT Scientific Publishing. 2014.
- [3] Rousseeuw, P. J.; Van Zomeren, B. C.: Unmasking Multivariate Outliers and Leverage Points. *Journal of the American Statistical Association* 85 (1990) 411, S. 633–639.
- [4] Gröll, L.: *Methodik zur Integration von Vorwissen in die Modellbildung*, Bd. 52. KIT Scientific Publishing. 2015.
- [5] Mikut, R.; Burmeister, O.; Reischl, M.; Loose, T.: Die MATLAB-Toolbox Gait-CAD. In: *Proc., 16. Workshop Computational Intelligence*, S. 114–124. Universitätsverlag Karlsruhe. 2006.
- [6] de Groen, P. P.: An Introduction to Total Least Squares. *arXiv preprint math/9805076* (1998).
- [7] Huber, P. J.: *Robust Statistics*. Springer. 2011.

- [8] Nelles, O.; Fischer, M.: Local Linear Model Trees (LOLIMOT) for Nonlinear System Identification of a Cooling Blast. In: *Proc., 4th European Congress on Intelligent Techniques and Soft Computing EUFIT'96*, S. 1187–1191. Aachen. 1996.
- [9] Rezaee, B.; Zarandi, M. F.: Data-Driven Fuzzy Modeling for Takagi–Sugeno–Kang Fuzzy System. *Information Sciences* 180 (2010) 2, S. 241–255.
- [10] Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer. 1995.
- [11] Vapnik, V. N.: *Statistical Learning Theory*. John Wiley. 1998.
- [12] Vapnik, V.; Golowich, S. E.; Smola, A.: Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing. In: *Advances in Neural Information Processing Systems 9*. Citeseer. 1996.
- [13] Smola, A.; Schölkopf, B.: A Tutorial on Support Vector Regression. *Statistics and Computing* 14 (2004) 3, S. 199–222.
- [14] Lauer, F.; Bloch, G.: Incorporating Prior Knowledge in Support Vector Regression. *Machine Learning* 70 (2008) 1, S. 89–118.
- [15] Bloch, G.; Lauer, F.; Colin, G.; Chamailard, Y.: Support Vector Regression from Simulation Data and Few Experimental Samples. *Information Sciences* 178 (2008) 20, S. 3813–3827.
- [16] Zhou, J.; Huang, J.: Incorporating Prior Knowledge Into Linear Programming Support Vector Regression. In: *International Conference on Intelligent Computing and Integrated Systems (ICISS)*, S. 591–595. IEEE. 2010.
- [17] Doneit, W.; Mikut, R.; Gröll, L.; Reischl, M.: Fragebogen zur Erfassung von Vorwissen in Funktionsapproximationen (Version 1.0). Techn. Ber., Institut für Angewandte Informatik, KIT. 2015. doi:10.13140/RG.2.1.3511.3446.
- [18] Doneit, W.; Tuga, M. R.; Mikut, R.; Liebetanz, D.; Rupp, R.; Reischl, M.: Kalibrierungs- und Trainingsstrategien zur individuellen Signalgenerierung für die myoelektrische Steuerung technischer Hilfsmittel. *Technisches Messen* 82 (2015) 9, S. 411–421.
- [19] Hoffmann, F.; Mikut, R.; Kroll, A.; Reischl, M.; Nelles, O.; Schulte, H.; Bertram, T.: Computational Intelligence: State-of-the-Art Methoden und Benchmarkprobleme. In: *Proc., 22. Workshop Computational Intelligence*, S. 15–29. KIT Scientific Publishing. 2012.

Inkrementelle nichtlineare Dimensionsreduktion

Jonas Schneider, Felix Igelbrink, Werner Brockmann

Universität Osnabrück
Albrechtstr. 28, 49069 Osnabrück

Tel.: (0541) 969 2439

Fax: (0541) 969 2799

E-Mail: {Jonas.Schneider, Felix.Igelbrink, Werner.Brockmann}@uos.de

Motivation

In den letzten Jahren hat der Einsatz von Methoden der Computational Intelligence in Bereichen mit hohen Datenmengen und hochdimensionalen Daten stark zugenommen. Die Einsatzbereiche sind dabei vielfältig und reichen von der automatisierten Auswertung großer wissenschaftlicher Experimente, über die Analyse und Vorhersage im kommerziellen oder medizinischen Bereich, bis zu Cyberphysikalischen Systemen und Big-Data- sowie Industrie 4.0-Anwendungen. Hochdimensionale Eingangsräume bereiten allerdings beim maschinellen Lernen große Probleme. So wächst mit steigender Dimensionalität die Größe des Eingangsraums sowie der Bedarf an Trainingsdaten exponentiell.

In vielen Einsatzbereichen ist es zusätzlich erforderlich, einen stetigen Strom neuer Daten zu verarbeiten, d.h., dass oft auch die Lerndaten inkrementell über der Zeit eintreffen. Zudem stellen Anwendungen wie die on-line und on-board Analyse von Daten, beispielsweise in Kraftfahrzeugen oder auf autonomen mobilen Robotern, bei der zerstörungsfreien Werkstoffprüfung oder bei der Analyse von multi- bzw. hyperspektralen Geodaten oft harte Echtzeitanforderungen. Die meisten der genannten Anwendungen laufen auf eingebetteten Systemen und es wäre vorteilhaft, wenn sie on-line lernen könnten, das heißt inkrementell eintreffende Lernstimuli zu verarbeiten. Hochdimensionale Probleme verursachen jedoch zusätzlich auch eine hohe Rechenzeit, sodass eine solche inkrementelle Verarbeitung oft nicht gewährleistet werden kann.

Um mit diesen Problemen umzugehen, bieten sich Methoden zur Dimensionsreduktion an, die ein hochdimensionales Problem in einen niedrigdimensionalen Raum transferieren. Das reduzierte Problem kann dann schneller und einfacher in diesem niedrigdimensionalen Raum gelöst werden.

Anforderungen

Ein lernfähiges System für hochdimensionale Daten sollte mit folgenden Problemen umgehen können:

- Große Eingangsräume führen zum sogenannten *Fluch der Dimensionalität* und damit mittelbar zu einer *dünnen Datenlage (Sparsity)*. Da diese Sparsity exponentiell mit der Anzahl Eingangsdimensionen wächst, müssen die lernfähigen Systeme durch Ausnutzen von Vorwissen in jeder Dimension genau an das jeweilige Lernproblem angepasst werden, um trotzdem schnell zu konvergieren. Dies ist jedoch für den Entwickler umso schwieriger zu entwerfen, je höher die Dimensionalität des Problems und damit seine Komplexität ist.
- Das Lernverfahren muss auf der anderen Seite *jedes Trainingsdatum optimal ausnutzen*, um die Lernzeit bzw. die benötigte Anzahl an Trainingsdaten möglichst gering zu halten, gleichzeitig jedoch eine unzulässige Extrapolation vermeiden. Das Sparsity-Problem muss somit explizit durch den Entwickler bereits zur Designzeit durch Einbringen des Vorwissens angegangen werden, um schwerwiegende Fehler, die die Sicherheit der Anwendung kompromittieren oder zumindest die Konvergenzgeschwindigkeit des Approximators verringern, zu vermeiden.
- Die Speicherung aller Trainingsdaten ist in on-line Szenarien unmöglich. Eine Speicherung einer Auswahl relevanter Trainingsdaten ist durch die benötigte Zeit zur Entscheidung, ob ein Trainingsdatum relevant ist, ebenfalls nur schwer möglich. Es ist somit sinnvoll, wenn auf einem stetigen Strom von Eingangsdaten gelernt wird, *Lerndaten einzeln zu betrachten und zu verarbeiten*. Das Lerngesetz muss somit jedes Trainingsdatum sofort so effektiv und so schnell wie möglich verarbeiten.
- Um einen sicheren und robusten Betrieb des Gesamtsystems gewährleisten zu können, müssen die Entwurfsentscheidungen und -parameter *transparent und interpretierbar* in ihrer Wirkung sein.

Viele reale Probleme wirken nur nach außen hochdimensional, besitzen intern aufgrund teilweiser Korrelationen zwischen den Eingangsgrößen jedoch eine niedrigdimensionale innere Struktur. Diese Struktur wird im Folgenden als *Innere Problemstruktur (Inner Problem Structure (IPS))* bezeichnet. Liegt einem Problem eine solche IPS zu Grunde, so kann die Komplexität des Problems für nachfolgende Algorithmen durch Ausnutzen der IPS vermindert werden, indem die Daten aus dem Eingangsraum in einen zur IPS passenden niedrigdimensionalen Transferraum (*Transfer-Space (T-Space)*) abgebildet werden. Bei einer solchen

Dimensionsreduktion ist besonders wichtig, dass die Nachbarschaftsbeziehungen der Datenpunkte in der IPS bei der Übertragung in den T-Space erhalten bleiben, damit dieser auch den Verlauf der IPS korrekt wiedergibt. Bei der inkrementellen Dimensionsreduktion ist zusätzlich ein *schnelles und sicheres Finden einer angemessenen Einbettung der IPS* in den T-Space essenziell.

Beispiele für Probleme, die eine solche IPS aufweisen, sind die Bestimmung von Konvektion in Wetterdaten [1], akustische Sensornetzwerke für die Quellenlokalisierung [2], 6D-Posenschätzer auf Basis von Bilddaten [3], Verarbeitung von 3D-Punktwolken [4] und Objekterkennung in Bilddaten [5] oder die Schätzung von Pflanzeigenschaften wie dem Chlorophyllgehalt aus hyperspektralen Geodaten [6]. Konkret werden beispielsweise im Bereich der Konvektionserkennung 22 Dimensionen auf 8 reduziert [1] und im Beispiel der akustischen Sensornetzwerke 49 Dimensionen auf 7 [2].

Stand der Technik

Lineare Dimensionsreduktionstechniken

Die einfachste Klasse der Dimensionsreduktionstechniken bilden lineare Verfahren wie die Independent Component Analysis (ICA), Principal Component Analysis (PCA), Multi-Dimensional Scaling (MDS) und Singular Vector Decomposition (SVD). Diese Techniken versuchen lineare Abhängigkeiten in einem niedrigdimensionalen Unterraum des Eingangsraums zu finden. Dabei muss der so ermittelte Unterraum die IPS des Problems wiedergeben. Obwohl diese Methoden effizient implementierbar sind und häufig angewendet werden, sind sie für viele Probleme ungeeignet, da sie nur lineare Abhängigkeiten erkennen können. Dies ist in Bild 1 an einem Beispiel einer Reduktion von zwei auf eine Dimension gezeigt. Es ist zu erkennen, dass der Ansatz (hier PCA) zur linearen Dimensionsreduktion die s-förmige IPS des Problems nicht angemessen erkennen und repräsentieren kann, während das nichtlineare Verfahren (hier eine Self-Organizing Map) die eindimensionale IPS korrekt findet.

Nichtlineare Dimensionsreduktionstechniken

Zur Approximation nichtlinearer IPS werden nichtlineare Dimensionsreduktionstechniken (engl. manifold learning) benötigt. Diese sind in der Lage bei hinreichend großer Ausdrucksstärke und einer hinreichenden Menge an Trainingsdaten eine nichtlineare Einbettung beliebig genau wiederzugeben. Die einfachsten

dieser Techniken gehen auf das klassische *Sammon's Mapping* [7] zurück. Die Idee dabei ist die Minimierung folgender Fehlerfunktion:

$$E = \frac{1}{\sum_{i < j} \dot{d}_{ij}} \sum_{i < j} \frac{(\dot{d}_{ij} - d_{ij})^2}{\dot{d}_{ij}} \quad (1)$$

Dabei gibt d_{ij} die Distanz zwischen dem i -ten und dem j -ten Datenpunkt im Eingangsraum und \dot{d}_{ij} die Distanz der Datenpunkte im T-Space an. Beispiele für darauf basierende Methoden sind die *Curvilinear Component Analysis* [8] und die *Curvilinear Distance Analysis* [9]. Da die Speicherung aller Punktdistanzen im Eingangsraum für diese Art von Ansätzen zwingend notwendig ist, eignen sie sich nicht für eine inkrementelle Verarbeitung.

Techniken wie *Isomap* [10], *Locally-Linear-Embedding* [11] und *Laplacian Eigenmaps* [12] nutzen auch Nachbarschaftsinformationen aller Datenpunkte. Sie erfordern somit auch mit einer steigenden Anzahl Trainingsdaten einen höheren Speicher- und Rechenaufwand und sind damit für inkrementelle Verarbeitung ebenfalls ungeeignet.

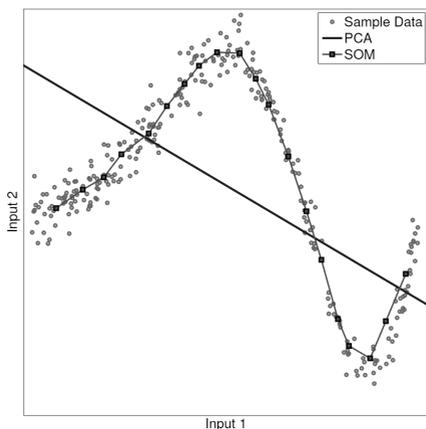


Bild 1: Vergleich einer linearen und einer nichtlinearen eindimensionalen Einbettung einer nichtlinearen IPS für ein zweidimensionales Problem. Als lineare Verfahren ist PCA dargestellt. Als nichtlineares Verfahren wird eine Self-Organizing Map mit 20 Knoten genutzt.

Kernelbasierte Methoden wie *Kernel-PCA* [13], eine nichtlineare Generalisierung der klassischen PCA, repräsentieren die Einbettung als Kernelstruktur, die aus allen Trainingsdaten generiert wird. Damit erfordern diese Methoden ebenfalls eine Speicherung aller bisherigen Trainingsdaten. Neue Ansätze begrenzen den dafür benötigten Speicher durch eine Auswahl repräsentativer Datenpunkte. In hochdimensionalen Eingangsräumen ist dieser Ansatz durch die Sparsity der Daten jedoch nicht sinnvoll. Ebenso ist die Auswahl der Trainingsdaten ein kritischer Faktor, der die Robustheit des Verfahrens stark kompromittieren kann und viel Rechenzeit kostet. Um die Auswahl sinnvoll treffen zu können, muss die Struktur der Daten dem Entwickler vorab bekannt sein. Andernfalls kann eine beschränkte Kernel-PCA nicht garantieren, die Einbettung überhaupt finden zu können.

Techniken, die ohne Speicherung der Eingangsdaten auskommen, sind künstliche neuronale Netze. Beispielsweise ist *Nonlinear PCA* [14] eine solche Methode, die mittels Backpropagation ein neuronales Netz so trainiert, dass es die Dimensionsreduktion durchführt. Die Lern- und die Auswertephase sind dabei getrennt. Nach dem Lernen werden dann nur die Eingänge des Netzes genutzt, die ein signifikantes Gewicht besitzen. Es handelt sich also um eine automatisierte Form der *Feature Extraction*. Dieser Ansatz ist sehr ressourcenintensiv und wächst in seiner Komplexität stark an, wenn die Ausdrucksfähigkeit, also die Anzahl Neuronen des Netzes, erhöht wird. Ein weiteres Problem liegt in der zeitlichen Trennung der Lern- und der Auswertephase. Hierdurch kann sich ein einmal trainiertes Netz nicht an zeitvariante Prozesse anpassen. Damit eignet sich dieser Ansatz nicht für jede inkrementelle Verarbeitung.

Ein weiterer Ansatz sind Neuronale Netze als *Autoencoder* [15], die eine hochdimensionale Eingabeschicht auf eine niedrigdimensionale versteckte Schicht abbilden und anschließend wieder eine hochdimensionale Ausgabeschicht besitzen. Es wird nun die Identitätsabbildung trainiert. Die versteckte Schicht entspricht dann einer niedrigdimensionalen Einbettung. Lernverfahren für Autoencoder sind jedoch ebenfalls nicht echtzeitfähig und inkrementell, also somit hier nur aufwendig einsetzbar.

Ein anderer neuronaler Ansatz sind die *Self-Organizing Maps (SOM)* [16]. Diese bilden den d -dimensionalen Eingangsraum auf ein im Vorfeld festgelegtes niedrigdimensionales regelmäßiges Gitter bestehend aus Knoten und Kanten ab. Dieses Gitter besteht aus k Knoten in jeder der inneren n Dimensionen und besitzt somit eine Gesamtanzahl von $K = k^n$ Knoten. Beim Lernen werden die Knotenpositionen typischerweise im Batch in Richtung der Daten bewegt.

Die entsprechende Vorschrift dafür lautet:

$$w_i^{t+1} = w_i^t + \epsilon^t \cdot h_{si}^t \cdot (x^t - w_i^t) \quad (2)$$

Dabei gibt $x^t \in \mathbb{R}^d$ die Position des Lerndatums im Eingangsraum zum Zeitpunkt t an, $w_i^t \in \mathbb{R}^d$ die Position des i -ten Knoten im Eingangsraum zum Zeitpunkt t , ϵ^t die zeitabhängige Lernrate und h_{si}^t eine Gewichtungsfunktion basierend auf der topologischen Distanz d_A mit:

$$h_{si}^t = e^{-\frac{d_A(k_s, k_i)^2}{2(\delta^t)^2}} \quad (3)$$

k_s und k_i geben dabei die Positionen der Knoten in der SOM-Topologie an. Die Lernrate bestimmt sich durch:

$$\epsilon^t = \epsilon_{start} \cdot \left(\frac{\epsilon_{end}}{\epsilon_{start}} \right)^{\frac{t}{\tau}}, \epsilon_{end} < \epsilon_{start} \quad (4)$$

Die zeitabhängige Gewichtungsfunktion h ist abhängig vom Adaptionradius δ , der gegeben ist durch:

$$\delta^t = \delta_{start} \cdot \left(\frac{\delta_{end}}{\delta_{start}} \right)^{\frac{t}{\tau}}, \delta_{end} < \delta_{start} \quad (5)$$

Es werden somit alle Knotenpositionen abhängig von ihrer topologischen Entfernung zu dem Knoten, der dem Eingangsdatum am nächsten liegt (*Best Matching Unit (BMU)*) angepasst. Dabei nehmen sowohl die Lernrate ϵ als auch der Adaptionradius δ über die Zeit von ihrem Startwert zu ihrem Endwert kontinuierlich ab.

Eine zur Dimensionsreduktion eingesetzte SOM bildet dabei nach Abschluss des Trainings jedes Eingangsdatum auf seine BMU ab. Die Gitterkoordinaten der BMU bilden dann die Ausgabe im niedrigdimensionalen T-Space. Dadurch wird der kontinuierliche Eingangsraum auf einen *diskreten* und *konstanten* T-Space abgebildet. In Bild 2 ist dieser Vorgang für ein einfaches 2D-Beispiel dargestellt.

Wenn eine kontinuierliche Ausgabe erforderlich ist, kann mittels der Aktivierungen der die BMU umgebenden Knoten eine Interpolation durchgeführt werden. Dazu existieren verschiedene Ansätze [17]. Wir konzentrieren uns im Folgenden allerdings auf die Dimensionsreduktion selbst und nicht auf die anschließende Auswertung.

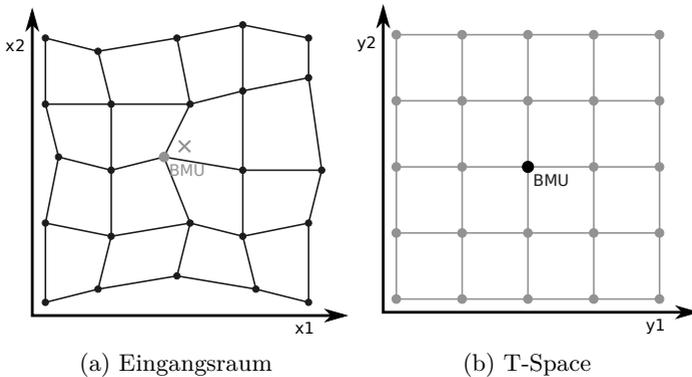


Bild 2: Eine SOM bildet einen Datenpunkt im Eingangsraum über seine BMU in den diskreten, konstanten T-Space ab, hier zur Veranschaulichung des Prinzips von 2D nach 2D gezeigt.

Klassischerweise sind SOMs Werkzeuge, die zur Visualisierung von Datensätzen eingesetzt werden. Dafür wird eine Dimensionsreduktion meist auf eine 2-dimensionale Struktur vorgenommen, um sie im Anschluss daran grafisch darzustellen. Um eine angemessene Auflösung zu erhalten, wird dabei eine hohe Anzahl Knoten (>100) für beide der Ausgangsdimensionen eingesetzt. Nebeneffekt hierdurch ist, dass solche SOMs zu Overfitting neigen und nicht in der Lage sind, Probleme mit höherdimensionalen IPS korrekt abzubilden.

Da SOMs einen konstanten Ressourcenbedarf haben und einfach ausgewertet und fortlaufend gelernt werden können, stellen sie die vielversprechendste Technik zur inkrementellen nichtlinearen Dimensionsreduktion unter Erfüllung der eingangs aufgeführten Anforderungen dar. Der SOM-Algorithmus muss dafür jedoch angepasst werden, um auch höhere IPS-Dimensionen bedienen zu können und Daten inkrementell statt als Batch verarbeiten zu können.

Konzept zu inkrementellen Dimensionsreduktion

Grundproblem

Das konkrete Ziel dieser Arbeit ist es, eine nichtlineare Dimensionsreduktion für hochdimensionale Eingangs- und Transferräume zu entwickeln, die ein inkremen-

telles Lernen auf einem eingehenden Datenstrom ermöglicht. Als Basisalgorithmus dient dazu die SOM.

Da die Datenpunkte einzeln verarbeitet werden müssen und ihre konkrete Verteilung sowie Reihenfolge unbekannt sind, ist das Optimieren der Parameter der SOM jedoch sehr aufwendig bis unmöglich. Diese Problematik betrifft insbesondere die Initialwerte des Adaptionradius und der Lernrate sowie ihr Abklingverhalten über die Zeit.

Eine nicht optimale Anpassung dieser Parameter führt insbesondere zu Problemen bei der Erfassung der globalen Struktur der Trainingsdaten. So neigen klassische SOMs zu dem in der Literatur bereits bekannten Tangling [18]. Bild 3 zeigt ein Beispiel für einen solchen Fall. Anhand der Nummerierung der Knoten ist zu erkennen, dass diese zwar auf den Trainingsdaten liegen, ihre globale Ordnung allerdings fehlerhaft ist und zu starken *Topologiedefekten* führt. Diese sind besonders dadurch zu erkennen, dass die Verbindungslinien benachbarter Knoten durch Bereiche ohne Datenpunkte verlaufen. Als Resultat sind manche im T-Space benachbarte Knoten im Eingangsraum weit voneinander entfernt. Die Nachbarschaftserhaltung zwischen Eingangsraum und T-Space ist bei einer solchen SOM somit nicht mehr gegeben und eine Auswertung basierend auf der topologischen Nachbarschaft (vgl. Interpolation) führt potenziell zu großen Fehlern. Um eine korrekte Auswertung sicherzustellen, ist es folglich notwendig, dass der T-Space, und damit die topologische Ordnung der SOM, hinreichend gut zur IPS der Trainingsdaten passt.

Unser Ansatz ist hier, das Tangling gezielt durch eine *Bootstrap-SOM*, einer Erweiterung der Standard-SOM, aufzulösen und gleichzeitig das Gitter der SOM mittels inkrementeller Eingriffe genauer an die IPS anzupassen. Dabei werden die auftretenden Topologiedefekte frühzeitig aktiv behoben. Das Konzept der Bootstrap-SOM wird im Folgenden detaillierter vorgestellt.

Bootstrap-SOM

Wie bereits im vorherigen Abschnitt erwähnt, ist es, aufgrund der Unbekanntheit der Datenverteilung zu Beginn des inkrementellen Lernens, sehr schwierig bis unmöglich die Parameter einer SOM zur Designzeit optimal einzustellen. Im Folgenden wird daher die Standard-SOM zur Bootstrap-SOM erweitert. Diese überwacht den Lernvorgang auf einer Metaebene und stellt den Adaptionradius dynamisch abhängig vom Entfaltungszustand des SOM-Gitters ein. Dies soll insbesondere Topologiedefekte frühzeitig aktiv beheben und erst danach die lokale Entfaltung des Gitters vornehmen. Die dazu verwendeten Einstellregeln sind:

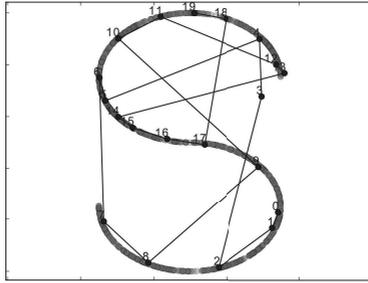


Bild 3: Tangling bei einer lokal trainierten 1D SOM.

- **Wenn** bei einem Trainingsdatum ein Topologiedefekt erkannt wird, **dann** wird der Adaptionradius gemäß Gleichung 6 erhöht um diesen Fehler zu beseitigen.

$$r(t + 1) = d_T + 1, \text{ falls } r(t) < d_T \quad (6)$$

Um einen Topologiedefekt zu beheben, muss der Adaptionradius $r(t)$ mindestens so groß eingestellt werden wie die topologische Distanz d_T der beiden fehlerverursachenden Knoten. Da typischerweise mehrere Schritte zur Eliminierung des Defektes erforderlich sind, wird der neue Radius $r(t + 1)$ bewusst etwas größer gewählt.

- **Wenn** für das Trainingsdatum kein Topologiedefekt erkannt wird, oder **wenn** $r(t) > d_T$ ist, **dann** klingt der Adaptionradius analog zur Standard-SOM automatisch mit konstanter Halbwertszeit ab.

Sobald der Adaptionradius auf einen vorher definierten Minimalwert abgefallen ist (hier 1) nimmt die Bootstrap-SOM für ein statisches Problem nur noch lokale Anpassungen vor. Erst dann ist das Gitter vollständig entfaltet und die Bootstrap-SOM liefert überall brauchbare Ergebnisse.

Durch die dynamische Anpassung des Adaptionradius treten zwei Qualitäten beim Lernen der IPS zutage, die bei der Bootstrap-SOM in Kombination behandelt werden. Die erste Qualität ist das *schnelle Finden einer passenden Einbettung in die IPS*. Das Konzept der Bootstrap-SOM zielt primär auf diese erste Qualität, da diese auch insbesondere die Beseitigung des Tanglings durch Umsortieren der Nachbarschaften umfasst. Die zweite Qualität ist dagegen das *fortlaufende Feintuning* der einmal gelernten Einbettung mittels graduellen Anpassungen der Knoten. Das Feintuning findet nach der Konvergenz der Bootstrap-SOM lokal

statt, das heißt, die Nachbarschaftsbeziehungen werden nicht mehr verändert. Da mit der Konvergenz der Bootstrap-SOM deren eigentliche Aufgabe abgeschlossen ist, bezeichnen wir das Training der Bootstrap-SOM auch als *Bootstrapping*, daher auch der Name „Bootstrap-SOM“.

Anders als beim Batch-Lernen spielt hier die Verteilung der Trainingsdaten im Eingangsraum eine entscheidende Rolle, ist jedoch nicht vorhersehbar. Daher muss ständig nachgelernt werden können. Wir verwenden aus diesem Grund bei der Bootstrap-SOM eine konstante Lernrate. Im Hinblick auf zeitliche Veränderungen des unterliegenden Prozesses kann es jedoch sinnvoll sein, auch die Lernrate mittels eigener Einstellregeln dynamisch anzupassen. Dies liegt jedoch hier nicht im Fokus.

Veranschaulichungsbeispiel

Zur beispielhaften Veranschaulichung der vorgestellten Bootstrap-SOM wird hier eine 2D-SOM mit 15 Knoten pro T-Space-Dimension genutzt, die eine quadratische Ebene als IPS approximieren soll. Die Erzeugungsvorschrift für diese Ebene lautet:

$$x = [x_1; x_2; 0.5], x_1, x_2 \in \mathcal{U}([0; 1]^2) \quad (7)$$

mit gleichverteilten Trainingsdaten. Diese sehr einfache IPS dient hier als anschauliches Beispiel, um die Probleme der Standard-SOM mit unterschiedlichen Datenverteilungen, insbesondere das Tangling, im Vergleich mit der Bootstrap-SOM zu zeigen.

Die Bootstrap-SOM wird mit einer Standard-SOM, deren Parameter manuell optimiert wurden, hinsichtlich verschiedener Metriken verglichen:

- Der Quantisierungsfehler (*Quantisation Error*) ist der mittlere Abstand der Trainingsdaten von ihrer BMU nach dem Abschluss des Trainings.
- Die Verbleibende Varianz (*Remaining Variance*) gibt den Abstand aller Datenpunkte zu ihren Projektionen auf die SOM an und wird durch die Division durch die Gesamtvarianz aller Datenpunkte normalisiert. Sie beschreibt zusammen mit dem Quantisierungsfehler wie gut die SOM die IPS wiedergibt.

- Der Topografische Fehler (*Topographic Error*) ist die prozentuale Anzahl der Trainingsdaten, bei denen Topologiedefekte erkannt werden. Er beschreibt den Grad der Topologieerhaltung der SOM zwischen Eingangsraum und T-Space.

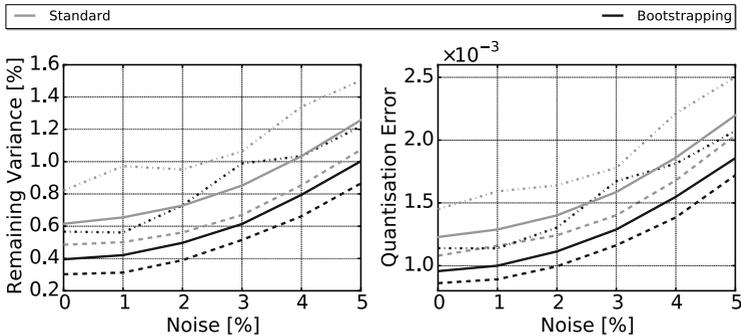
Beide Verfahren werden für unterschiedliche Rauschniveaus auf den gleichen Daten ausgeführt. Um die Untersuchung statistisch abzusichern, werden dabei jeweils 100 Durchläufe mit unterschiedlichen Datenverteilungen durchgeführt und die Metriken darüber gemittelt. Um die beiden Ansätze vergleichen zu können, zeigen die Ergebnisse in Bild 4 die minimalen, maximalen sowie die gemittelten Metriken aller 100 Durchläufe nach 1500 inkrementellen Trainingsschritten. Für den Topografischen Fehler sind die prozentualen Anteile verschiedener Fehlermagnituden über alle Durchläufe abgebildet.

Es ist zu erkennen, dass die Bootstrap-SOM im Vergleich zur Standard-SOM in allen Metriken im Mittel kleinere Fehler liefert, obwohl keine Handoptimierung erfolgte. Dies ist insbesondere darauf zurückzuführen, dass es bei der Bootstrap-SOM wegen der dynamischen Anpassung des Adaptionradius keine Fälle von bleibenden topologischen Fehlern in der Struktur des T-Space gibt. Einzig in den Fällen, in denen die Parameter einer Standard-SOM exakt auf die zufällige Datenlage und Datenverteilung passten, liefert diese bessere Ergebnisse als die Bootstrap-SOM. In einem inkrementellen Einsatzszenario sind jedoch weder die Daten noch ihre Verteilung vorab exakt bekannt, sodass eine solche perfekt angepasste Standard-SOM nur für vorher bekannte synthetische Daten existiert.

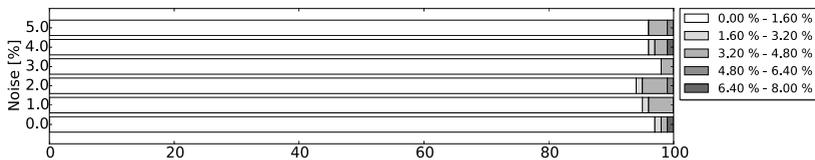
Von besonderem Interesse ist hier der Topografische Fehler. Aus Bild 4b ist zu entnehmen, dass für diese sehr einfache IPS selbst bei nicht verrauschten Daten einige Durchläufe der Standard-SOM zu Einbettungen führen, bei denen bis zu 8% der Trainingsdaten zu Topologiedefekten führen, was sich in schweren Strukturdefekten äußert. In Bild 5 ist ein solcher Fall exemplarisch für ein Rauschen von 4% für beide SOM-Varianten gegenübergestellt. Bei der Bootstrap-SOM treten diese großen Fehler aufgrund der dynamischen Anpassung des Adaptionradius nicht mehr auf. Erst bei einem Rauschniveau von 5% treten in einigen Durchläufen größere Fehler auf (weniger als 3,2% der Trainingsdaten).

Diskussion

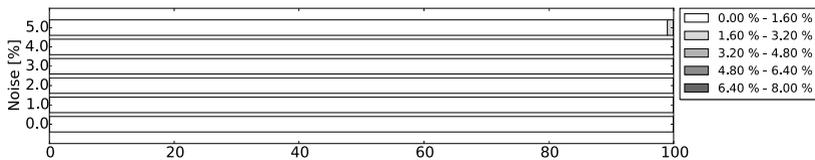
Der in diesem Beitrag vorgestellte Ansatz zur inkrementellen Dimensionsreduktion erlaubt es, on-line eine niedrigdimensionale innere Struktur auf einem eingehenden Datenstrom zu lernen. Um das vorrangige Ziel, nämlich das korrekte Extrahieren der Nachbarschaftsbeziehungen zwischen IPS und T-Space aus den



(a) Links: Verbleibende Varianz nach 1500 Lernschritten. Rechts: Quantisierungsfehler für die gleichen Evaluationen. Die gepunkteten Kurven geben das Maximum über alle 100 Durchläufe, die durchgezogenen Linien den Mittelwert über alle Durchläufe und die gestrichelten Linien das Minimum über alle Durchläufe an.



(b) Topographischer Fehler bei der Standard-SOM.



(c) Topographischer Fehler bei der Bootstrap-SOM.

Bild 4: Vergleich der drei Metriken auf dem Surface-Datensatz.

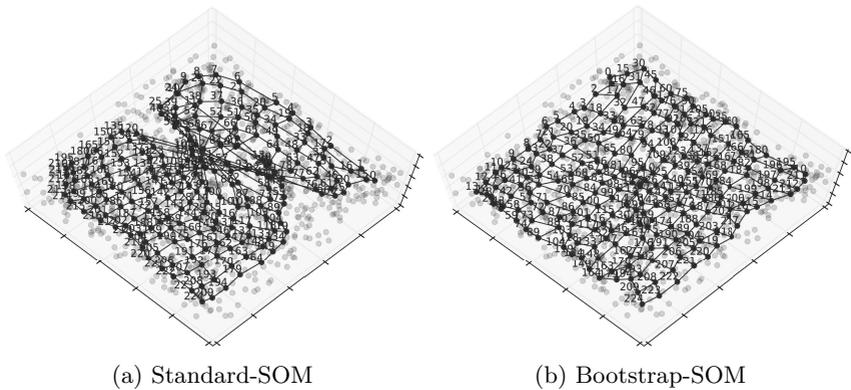


Bild 5: Gefundene Einbettungen der Standard-SOM und der Bootstrap-SOM im Vergleich.

Trainingsdaten zu erreichen, müssen sowohl für die Standard- als auch für die Bootstrap-SOM die Dimensionalität sowie die Ausdrucksstärke des T-Space in den einzelnen Dimensionen zum unterliegenden Problem passend gewählt werden. Hierzu benötigt der Entwickler ein grobes Wissen über die IPS des Problems. Eine ungünstige Spezifizierung der SOM-Struktur für das unterliegende Problem führt schlimmstenfalls zu einer starken zusätzlichen Faltung des SOM-Gitters in die IPS (Tangling) oder die SOM ist gar nicht in der Lage die IPS korrekt zu erkennen.

Weiterhin müssen in allen Bereichen des Eingangsraumes, die im späteren Betrieb abgedeckt werden müssen, hinreichend viele Trainingsdaten vorliegen, da komplett unbekannte Gebiete von einer SOM grundsätzlich nicht erfasst werden können. Für die Bootstrap-SOM kommt erschwerend hinzu, dass diese in derartigen Gebieten nicht in der Lage ist topologische Defekte zu beseitigen, da deren Erkennung mithilfe der Eingangsdaten geschieht.

Standard- und Bootstrap-SOM kommen ohne Speicherung alter Trainingsdaten aus. Dadurch skaliert der Speicherbedarf nur hinsichtlich der Größe der genutzten SOM, nicht jedoch hinsichtlich der Menge der verarbeiteten Trainingsdaten. Dies hat den Vorteil, dass sich dieser Ansatz somit für eine inkrementelle Verarbeitung auf beliebigen Datenströmen eignet.

Da durch die Lokalisierung der Knoten der SOM im Eingangsraum ein fester Zusammenhang zum T-Space besteht, ist dieser interpretierbar und es können bei korrekter Entfaltung Aussagen über die Beziehung zwischen IPS und T-Space getroffen werden.

Die Bootstrap-SOM besitzt den Vorteil, dass sich der Entwurfsaufwand für eine gut funktionierende SOM durch die automatische Einstellung des Adaptionradius gegenüber der Standard-SOM signifikant verringert. Zusätzlich werden die beiden Qualitäten des Findens der Einbettung sowie des Feintunings in Kombination von einem Algorithmus behandelt und erst zum Feintuning übergegangen, wenn das Finden der Einbettung abgeschlossen ist. Bei der Standard-SOM sind beide Qualitäten hingegen durch das Abklingen des Adaptionradius und der Lernrate strikt getrennt und es wird zu einem festen, vordefinierten Zeitpunkt zum Feintuning übergegangen. Dies ermöglicht der Bootstrap-SOM, besonders in einem inkrementellen Szenario flexibel auf unterschiedliche Datenverteilungen zu reagieren. Sowohl durch den verringerten Entwurfsaufwand als auch durch diese Flexibilität wird die Sicherheit eine gute Einbettung der IPS zu finden gegenüber der Standard-SOM deutlich erhöht.

Aufgrund der Abhängigkeit der Bootstrap-SOM von der Reihenfolge der Trainingsdaten ist jedoch nicht a-priori klar, wann die Bootstrap-SOM vollständig entfaltet ist. Bei sehr großen Eingangsräumen, einer hohen Zeitvarianz des unterliegenden Prozesses oder einer sehr ungünstigen Datenverteilung kann es passieren, dass die Bootstrap-SOM nicht hinreichend genug konvergiert, da neue Trainingsdaten immer wieder zu Topologiedefekten und damit zu einem höheren Adaptionradius führen.

Zusammenfassung

In diesem Beitrag ist gezeigt worden, dass hochdimensionale Eingangsräume und die resultierende dünne Datenlage für viele inkrementelle Lernverfahren eine große Herausforderung darstellen. Durch Ausnutzen der vielen Anwendungen unterliegenden IPS mittels einer nichtlinearen Dimensionsreduktion lässt sich die Komplexität nachfolgender Algorithmen stark verringern.

Es wurde gezeigt, dass sich viele der gängigsten Verfahren zur Dimensionsreduktion nicht für eine inkrementelle, datenstrombasierte Verarbeitung von Eingangsdaten eignen. Um eine solche inkrementelle Verarbeitung bei wechselnder Datenverteilung mit schneller Konvergenz auf die IPS zu ermöglichen, ist die Bootstrap-SOM als Ansatz zur schnellen Erfassung der globalen Topologie vorgestellt worden. Der klassischerweise manuell auf das Problem zu optimie-

rende Adaptionradius wird dabei durch eine Überwachung des Lernfortschritts automatisch eingestellt. In den Untersuchungen ist gezeigt worden, dass die Bootstrap-SOM die Probleme einer Standard-SOM in einem inkrementellen Szenario kompensiert und gleichzeitig den Tuning-Aufwand signifikant verringert.

Im Vergleich zu Verfahren, die viele Trainingsdaten speichern, ist die Bootstrap-SOM in der Lage auch inkrementell flexibel auf unterschiedliche Datenverteilungen zu reagieren und konvergiert schneller und sicherer als eine Standard-SOM auf die IPS. Gleichzeitig wird der Optimierungsaufwand gegenüber einer klassischen SOM deutlich verringert. Damit eröffnet die Bootstrap-SOM interessante Perspektiven für eine inkrementelle nichtlineare Dimensionsreduktion, die in Zukunft weiter ausgeführt werden, z.B. hinsichtlich der dynamischen Einstellung der Lernrate oder dem Einsatz für zeitvariante Prozesse.

Literatur

- [1] Zhiying, L.; Yuanxun, Z.; Hongmin, M.: Detection of strong convective weather based on manifold learning. In: *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, S. 1504–1508. IEEE. 2012.
- [2] Patwari, N.; Hero III, A. O.: Manifold learning algorithms for localization in wireless sensor networks. In: *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, Bd. 3, S. iii–857. IEEE. 2004.
- [3] Gashler, M.; Martinez, T.: Temporal nonlinear dimensionality reduction. In: *Neural Networks (IJCNN), The 2011 International Joint Conference on*, S. 1959–1966. IEEE. 2011.
- [4] Brunskill, E.; Roy, N.: SLAM using incremental probabilistic PCA and dimensionality reduction. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, S. 342–347. IEEE. 2005.
- [5] Seung, H. S.; Lee, D. D.: The manifold ways of perception. *Science* 290 (2000) 5500, S. 2268–2269.
- [6] Haboudane, D.; Tremblay, N.; Miller, J. R.; Vigneault, P.: Remote estimation of crop chlorophyll content using spectral indices derived from hyperspectral data. *Geoscience and Remote Sensing, IEEE Transactions on* 46 (2008) 2, S. 423–437.

- [7] Sammon, J. W.: A nonlinear mapping for data structure analysis. *IEEE Transactions on computers* (1969) 5, S. 401–409.
- [8] Demartines, P.; Héroult, J.: Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *Neural Networks, IEEE Transactions on* 8 (1997) 1, S. 148–154.
- [9] Lee, J. A.; Lendasse, A.; Verleysen, M.: Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing* 57 (2004), S. 49–76.
- [10] Tenenbaum, J. B.; De Silva, V.; Langford, J. C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290 (2000) 5500, S. 2319–2323.
- [11] Roweis, S. T.; Saul, L. K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* 290 (2000) 5500, S. 2323–2326.
- [12] Belkin, M.; Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation* 15 (2003) 6, S. 1373–1396.
- [13] Zhou, S.: Probabilistic analysis of kernel principal components: mixture modeling and classification. *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence* (2003).
- [14] Kramer, M. A.: Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal* 37 (1991) 2, S. 233–243.
- [15] Hinton, G. E.; Salakhutdinov, R. R.: Reducing the dimensionality of data with neural networks. *Science* 313 (2006) 5786, S. 504–507.
- [16] Kohonen, T.: The self-organizing map. *Neurocomputing* 21 (1998) 1, S. 1–6.
- [17] Göppert, J.; Rosenstiel, W.: Topology preserving interpolation in self-organizing maps. In: *Proceedings of the NeuroNIMES*, Bd. 93, S. 425–434. 1993.
- [18] Van Hulle, M. M.: Globally-ordered topology-preserving maps achieved with a learning rule performing local weight updates only. In: *Neural Networks for Signal Processing V. Proceedings of the 1995 IEEE Workshop*, S. 95–104. IEEE. 1995.

Approximatoren für On-line Regression

Jan H. Schoenke, Werner Brockmann

Universität Osnabrück
Albrechtstr. 28, 49069 Osnabrück

Tel.: (0541) 969 2439

Fax: (0541) 969 2799

E-Mail: {Jan.Schoenke, Werner.Brockmann}@uni-osnabrueck.de

Einführung

Im Bereich des Maschinellen Lernens bietet die inkrementelle oder On-line Regression effektive Methoden für Anwendungen, die von der adaptiven Modellierung eingebetteter Systeme über Regelungsaufgaben bis zur Analyse von Datenströmen reichen. Die inkrementelle Regression als überwachtetes Lernproblem wird üblicherweise als Kombination aus einem Approximator und einem Lernverfahren realisiert, die zusammen ein lernfähiges System bilden. Der Approximator stellt dabei die Wissensbasis dar, welche zum Beispiel das zu lernende Modell repräsentiert. Das Lernverfahren verarbeitet die inkrementell anfallenden Lerndaten und integriert diese in die bestehende Wissensbasis, um so die Vorhersage des Modells zu verbessern.

Beim Entwurf eines solchen lernfähigen Systems muss der Designer dessen einzelne Teile, den Approximator und das Lernverfahren, für die Anwendung passend auslegen, damit das lernfähige System dort sinnvoll eingesetzt werden kann und zum Beispiel schnell und sicher lernt. Darüber hinaus muss jedoch auch die Interaktion zwischen Approximator und Lernverfahren berücksichtigt werden, damit das Lernen überhaupt erfolgreich verlaufen kann und einen Mehrwert für die Anwendung bietet.

Im Rahmen dieser Arbeit werden formale Eigenschaften des Approximators betrachtet, die für den Erfolg des inkrementellen Lernens im Rahmen der On-line Regression relevant sind. Dazu werden verschiedene Klassen von Approximatoren hinsichtlich ihrer Eignung zur On-line Regression vergleichend auf einer mathematisch-methodischen Ebene diskutiert. Die aus der Theorie hergeleiteten Einsichten werden durch vergleichende Untersuchungen an ausgewählten Beispielen validiert. Für die Untersuchungen werden *Passive-Aggressive* (PA),

Incremental Risk Minimization Algorithm (IRMA), *Recursive Least Squares* (RLS) und *Second order Incremental Risk Minimization Algorithm* (SIRMA) als Vertreter für Lernverfahren 1. und 2. Ordnung verwendet. Der Fokus der Untersuchungen liegt dabei stets auf den Eigenschaften des verwendeten Approximators.

On-line Regression

Das Ziel jeder Regression ist es einen funktionalen Zusammenhang $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ zu finden, der die Relation zwischen einer Menge D von Datenpaaren (x, y) beschreibt. Für ein gutes Ergebnis der Regression gilt $f(x) \approx y$ unabhängig davon ob die Eingaben x aus der Menge D der Beispieldaten stammen oder nicht. Für den Spezialfall der On-line Regression steht die Menge D der Beispieldaten zu keinem Zeitpunkt vollständig zur Verfügung, sondern es werden inkrementell, meist in sukzessiven Zeitschritten, neue Beispieldaten für die Regression zur Verfügung gestellt.

Die On-line Regression arbeitet somit auf einem Strom von Daten (x_t, y_t) , der im Prinzip unendlich lang sein kann. Alle Daten zu sammeln und als Menge von Beispielen zu behandeln führt daher früher oder später zu Problemen hinsichtlich Rechenaufwand und Speicherbedarf.

Diesem Problem kann man prinzipiell auf zwei Weisen begegnen. Zum Einen kann man aus dem Datenstrom nur bestimmte Beispiele auswählen und speichern und diese für die weitere Verarbeitung verwenden. Zum Anderen kann man den gesuchten Zusammenhang durch eine bestimmte Klasse von Funktionen $f_\alpha(x)$ modellieren und die Beispieldaten dazu nutzen die Parameter des Modells α Datum für Datum einzustellen. Im Rahmen dieser Arbeit beschränken wir uns auf die Modellierung durch bestimmte Klassen von Funktionen $f_\alpha(x)$ und dem damit verbundenen Lernen der Modellparameter α . Für das parametrisierte Modell $f_{\alpha_t}(x)$ zum Zeitpunkt t mit dem Parametervektor α_t schreiben wir kurz $f_t(x)$. Die allgemeine Struktur der Modelle, die in dieser Arbeit betrachtet werden ist *Linear In den Parametern* (LIP), wie in Gleichung (1) beschrieben. Die Basisfunktionen ϕ_i des Modells bzw. des Approximators bestimmen dessen Klasse und Struktur. Sie vermitteln die Abbildung vom niederdimensionalen Eingangsraum in den höherdimensionalen Parameterraum. Die Eigenschaften dieser Abbildung sind essenziell für den Erfolg des Lernen.

$$f_\alpha(x) = \alpha^T \phi(x) = \sum_{i=1}^k \alpha_i \phi_i(x) \quad (1)$$

Der typische Ablauf eines Schrittes bei der On-line Regression verläuft wie folgt. Im Zeitschritt t liefert der Datenstrom das Beispiel (x_t, y_t) . Von diesem Datenpaar wird zunächst die Eingabe x_t verwendet, um die Vorhersage $\hat{y}_t = f_t(x_t)$ des aktuellen Modells f_t zu bilden. Mit dem Datenpaar (x_t, y_t) und der Vorhersage \hat{y}_t zusammen stellt anschließend ein Lernalgorithmus den Parametervektor des Modells α_t so ein, dass der aktuelle Fehler der Vorhersage $e_t = y_t - \hat{y}_t$ verringert wird. So kann die Vorhersage des Modells \hat{y}_t schrittweise verbessert werden.

Bei diesem Vorgehen ist der Lernalgorithmus mit verschiedenen Herausforderungen konfrontiert. So können zum Beispiel die Lernvorgaben y_t verrauscht sein und somit nicht den exakten gesuchten Zusammenhang beschreiben oder der gesuchte Zusammenhang verändert sich mit der Zeit, zum Beispiel durch Drift. Diese und weitere anwendungsspezifische Eigenschaften der Lernvorgaben y_t sind nicht im Fokus dieser Arbeit. Hier konzentrieren wir uns auf die Interaktion zwischen dem Lernalgorithmus und dem gewählten Modell $f_t(x)$ und dabei vor allem auf die Eigenschaften des Modells $f_t(x)$ und wie diese den Verlauf des On-line Lernens beeinflussen.

Die Eigenschaften des Modells sind von fundamentaler Bedeutung für das Lernen, da bei der On-line Regression in jedem Schritt nur genau ein Lerndatum (x_t, y_t) verarbeitet wird. Die neu hinzugewonnenen Informationen über die gesuchte Funktion f geben dabei nur Auskunft über einen einzigen Punkt im Eingangsraum, was dem Extremfall von maximaler Lokalität der Information und deren Geltungsbereich entspricht. Gleichzeitig ist es bei Modellen mit einer endlichen Anzahl an Parametern unmöglich nur für einen einzigen Punkt des Eingangsraums die Ausgabe zu verändern und die restliche Ausgabe unverändert zu lassen. Die Ausgabe sollte bei der Integration von neuen Lerndaten in das Modell nur in einem möglichst kleinen Bereich um das Lerndatum verändert werden und das andere schon vorhandene Wissen im Modell muss geschützt werden. Nur so lässt sich bei der Integration von neuen Lerndaten das Vergessen von altem Wissen verhindern. Dieses Problem entspricht einer Ausprägung des Reproduktions-Generalisierungs-Dilemmas für On-line Regression⁷.

Die Eigenschaft der On-line Regression in jedem Schritt nur ein Lerndatum zu verarbeiten macht die Anpassung der Parameter des Approximators im Sinne der linearen Algebra sehr schwierig, da ein stark unterbestimmtes lineares Gleichungssystem zu lösen ist. On-line Lernalgorithmen lösen dieses Problem mit unterschiedlichen Ansätzen wie im folgenden Kapitel dargestellt wird. Die

⁷ Für einen Lernschritt mehrere Lerndaten zu speichern und gleichzeitig zu verarbeiten kann die Situation etwas entspannen, ändert jedoch nichts an der Lokalität jedes einzelnen Lerndatums.

Beschränkung auf nur ein einziges Lerndatum ist jedoch auch die größte Stärke der On-line Regression, da so auch zeitvariante Funktionen gelernt werden können, indem sich das Lernen ständig an die Lerndaten und die dadurch repräsentierte Funktion adaptiert. Die Konvergenz hin zu einer bestimmten statischen Zielfunktion ist somit nur ein mögliches Ergebnis einer On-line Regression.

Um die Qualität einer On-line Regression zu bemessen ist daher der mittlere quadratische Fehler zu einer gewissen Zielfunktion weniger interessant als der *Cumulative Squared Loss* (CSL), also der kumulierte quadratische Vorhersagefehler, wie er in Gleichung (2) definiert ist. Diese skalare Größe über dem Verlauf des Lernens zu betrachten zeigt nicht nur ob zu einem bestimmten Zeitpunkt und wie schnell Konvergenz eintritt, sondern auch wie der Weg dorthin aussieht und das aus der Perspektive der Anwendung in der das lernfähige System eingesetzt wird. Die Untersuchungen im Rahmen dieser Arbeit werden daher mit dem CSL als Qualitätsmaß durchgeführt, auch wenn Zeitvarianz hier nicht explizit betrachtet wird.

$$CSL(T) = \sum_{t=1}^T e_t^2 = \sum_{t=1}^T (y_t - \hat{y}_t)^2 \quad (2)$$

On-line Lernverfahren

Im Folgenden werden kurz die Lernverfahren vorgestellt, die im Rahmen dieser Arbeit genutzt werden, um die Eigenschaften verschiedener Klassen von Modellen zu untersuchen. In der Literatur wird zwischen Lernverfahren 1. und 2. Ordnung unterschieden, je nachdem ob ein Lernverfahren direkt und ausschließlich auf den Lerndaten arbeitet (1. Ordnung) oder intern auch höhere Terme des Fehlerpotentials schätzt (2. Ordnung) auf dem das Lernverfahren operiert. Aus beiden dieser Kategorien werden jeweils zwei Lernverfahren vorgestellt, die je eine Klasse von Algorithmen ihrer Kategorie repräsentieren.

Lernverfahren 1. Ordnung

Als Vertreter von Lernverfahren 1. Ordnung werden hier die Algorithmen *Passive-Aggressive* (PA) [1] und *Incremental Risk Minimization Algorithm* (IRMA) [2] verwendet. In [1] werden mehrere Varianten von PA eingeführt von denen hier die Variante PA-II beschrieben wird, da diese einen direkten Vergleich mit IRMA zulässt.

PA-II führt das Parameterupdate gemäß Gleichung (3) aus, was einen Gradientenabstieg auf einem quadratischen Fehlerpotential mit zwei Komponenten entspricht. Der erste, passive Anteil bestraft Veränderungen am Parametervektor α , um vorhandenes Wissen zu schützen. Der zweite, aggressive Anteil bestraft Fehler in der Vorhersage und sorgt somit für die Integration von neuem Wissen durch die Lerndaten (x_t, y_t) . Insgesamt führt PA die Anpassung des Parametervektors so durch, dass für eine minimale Anpassung des Parametervektors α eine maximale Reduzierung des Fehlers e_t der Vorhersage erreicht wird.

$$\alpha_{t+1} = \phi(x_t) \frac{e_t}{\frac{1}{2c} + \phi(x_t)^T \phi(x_t)} \quad (3)$$

Die Grundidee von IRMA basiert ebenfalls auf einem Gradientenabstieg. Allerdings wird bei IRMA nicht die Veränderung des Parametervektors α bestraft, sondern die Veränderung der Ausgabe des Approximators f_t über dem kompletten Eingangsraum. So überträgt IRMA die Passivität von PA vom Parameterraum in den Eingangsraum. Dadurch wird nicht nur der Einfluss jedes einzelnen Parameters auf die Ausgabe beim Lernen berücksichtigt. Es wird darüber hinaus auch eine Lokalität der Auswirkungen des Parameterupdates erreicht, da Veränderungen in großer Entfernung zur aktuellen Eingabe x_t nicht dazu beitragen den Fehler zur Lernvorgabe y_t zu minimieren. Die Vorschrift zum Parameterupdate nach IRMA ist in Gleichung (4) angegeben. Dabei vermittelt die Matrix A den Einfluss der einzelnen Parameter untereinander und in Bezug auf die Ausgabe. Das ermöglicht es dem Algorithmus von einem bestimmten Approximator zu abstrahieren und verschiedene Approximatoren auch aus unterschiedlichen Klassen in ähnlicher Art und Weise zu behandeln.

$$\alpha_{t+1} = \alpha_t + A^{-1} \phi(x_t) \frac{y_t - \hat{y}_t}{\sigma + \phi(x_t)^T A^{-1} \phi(x_t)} \quad (4)$$

$$(A)_{i,j} = \int_D \phi_i(x) \phi_j(x) dx \quad (5)$$

Lernverfahren 2. Ordnung

Die Gegenstücke zu PA und IRMA in der Kategorie der Lernverfahren 2. Ordnung sind RLS [3] und SIRMA [2]. RLS minimiert rekursiv den mittleren quadratischen Fehler der Ausgabe des Approximators zu allen bisher verarbeiteten Lerndaten (x_t, y_t) . Dazu aggregiert RLS in einer Autokorrelationsmatrix S die jeweiligen Eingaben x_t der Lerndaten und kann so neu eingehende Lerndaten korrekt

gewichten. Um auf lange Sicht den mittleren quadratischen Fehler zu allen Lerndaten zu garantieren, erzwingt RLS durch ein Abklingen der Matrix S die Konvergenz des Parametervektors α . Mittelbar erzeugt RLS so eine Passivität des Parametervektors α , die vergleichbar mit dem Verhalten von PA ist.

Bei SIRMA wird die Grundidee von IRMA um eine on-line Dichteschätzung der Eingaben x_t ergänzt. Diese Dichteschätzung erlaubt es dem Algorithmus zu unterscheiden in welchen Regionen des Eingaberaums schon viele Lerndaten präsentiert wurden und wo nicht. Die Anpassung der Ausgabe wird in Regionen mit vielen Lerndaten reduziert, um eine durch die Lerndatendichte gewichtete Anpassung zu erhalten. Das resultierende Verhalten von SIRMA ähnelt somit dem von RLS, berücksichtigt jedoch zusätzlich die Passivität der Ausgabe des Approximators f im Eingangsraum.

Approximatoren

Vorbemerkungen

Im folgenden werden die verschiedenen Klassen von Approximatoren aus mathematisch-methodischer Sicht vorgestellt, die im Rahmen dieser Arbeit betrachtet werden. Jede einzelne dieser Klassen repräsentiert einen bestimmten Typ von universellem Funktionsapproximator und garantiert somit bei hinreichender Ausdrucksstärke eine bestimmte Approximationsgüte. Die Unterschiede im Approximationsverhalten der hier gewählten Klassen zeigen sich daher gerade bei begrenzter Ausdrucksstärke durch begrenzte Ressourcen, wie es in realen Anwendungen der Fall ist. Die hier getroffene Auswahl beschränkt sich auf reguläre oder regelmäßige Strukturen, da sich diese effizient durchsuchen lassen. Außerdem werden nur Approximatoren betrachtet, die LIP sind, da sich so zur Laufzeit ein einfaches Optimierungsproblem für die Parameteranpassung ergibt.

Bei der Vorstellung der Klassen von Approximatoren wird neben deren Konzept auch auf ihre Approximationseigenschaften eingegangen, die sich allgemein formalisiert betrachten lassen.

Die zentrale formale Eigenschaft, die hier betrachtet wird, lässt sich aus dem Vergleich der Updatevorschriften von PA-II und IRMA herleiten. Der wesentliche Unterschied zwischen den beiden liegt in der Matrix A , welche die *Gram-Matrix* [5] des jeweiligen Approximators f_t ist. Wird diese Matrix A zur Identität, so verhalten sich die beiden Lernverfahren identisch. Dies ist genau dann der Fall, wenn die Basisfunktionen ϕ_i des Approximators ein Orthonormalsystem bilden.

Die Eigenschaften der Basis des Funktionenraumes, die von den Basisfunktionen ϕ_i eines Approximators f erzeugt wird, hat somit unmittelbare Auswirkungen auf das Lernen. Um die Ähnlichkeit der Gram-Matrix A eines beliebigen Approximators zur Identität zu bemessen, wird hier die *Konditionszahl* der Gram-Matrix A betrachtet. Es gilt $\text{Cond}(A) = 1$ genau dann, wenn $A = I$ gilt. Die Konditionszahl wird umso größer je unähnlicher die Gram-Matrix A der Identität wird. Auf Grundlage der formalen Beobachtungen zur Kondition eines Approximators werden hier auch Erwartungen an die Ergebnisse der Untersuchungen formuliert, die im weiteren Verlauf überprüft werden.

Eine weitere formale Eigenschaft von Approximatoren, die in der Literatur diskutiert wird, ist die Lokalität Λ der Basisfunktionen ϕ_i gemessen in deren *support*, siehe Gleichung (6) und zum Beispiel [4].

$$\Lambda(\phi_i) = \text{supp}(\phi_i) = \overline{\{x \in \mathbb{R}^n | \phi_i(x) \neq 0\}} \quad (6)$$

Lokale Approximatoren haben Basisfunktionen ϕ_i deren Lokalität Λ stets deutlich kleiner ist als der gesamte Eingangsraum und unterstützen so das Lernverfahren dabei die lokal gültigen Lerndaten durch ausschließlich lokale Veränderungen am Approximator in diesen zu integrieren. *Globale Approximatoren* besitzen nur oder zum größten Teil Basisfunktionen ϕ_i deren *support* sich über den gesamten Eingangsraum erstreckt und es für das Lernverfahren sehr schwer machen nur lokale Veränderungen am Approximator vorzunehmen.

Polynome

Eine weit vorbereitete Klasse von globalen Funktionsapproximatoren sind Polynome [5]. Für die hier angestrebten Betrachtungen sind sie von besonderem Interesse, da für den Funktionenraum der Polynome verschiedene Basen in der Literatur diskutiert und eingesetzt werden. Diese Basen bzw. Repräsentationen von Polynomen sind beispielsweise Monome [5], Legendre- [4] und Chebyshev-Polynome [6].

Als weitere Basis seien hier Lagrange-Polynome [7] erwähnt, die auf einer Menge von Punkten definiert sind. Deren Anzahl legt mittelbar den Grad der verwendeten Polynome fest. Dabei bestimmt die Verteilung der zur Approximation gewählten (Basis-)Punkte die Eigenschaften der einzelnen Basisfunktionen ϕ_i und führt in gewisser Weise zu einer Lokalisierung der Basisfunktionen an den ihnen zugehörigen Punkten x_i .

Im Rahmen dieser Arbeit können nicht alle hier aufgeführten Basen für Polynome untersucht werden, da allein die Verteilung der Punkte bei Lagrange-Polynomen deren Eigenschaften so stark beeinflusst, dass es hierzu eigene Literatur gibt. Wir beschränken uns daher an dieser Stelle auf den Vergleich von Monomen und Legendre-Polynomen als Basis, da diese sich hinsichtlich ihrer Gram-Matrix A signifikant unterscheiden und so auch ein stark unterschiedliches Approximationsverhalten erwarten lassen. Die Unterschiede zwischen Legendre- und Chebyshev-Polynomen sind geringer als zwischen Legendre- und monomialen Polynomen, weshalb hier ein sehr ähnliches Approximationsverhalten zu erwarten ist.

Die Gram-Matrix von Polynomen in monomialer Basis verhält sich im Allgemeinen wie die Hilbert-Matrix, welche sehr schlechte numerische Eigenschaften hat wie zum Beispiel eine extrem hohe Konditionszahl. Die Gram-Matrix von Legendre-Polynomen ist die Identität, was bedeutet, dass die Basisfunktionen ϕ_i eine orthonormale Basis bilden. Im direkten Vergleich ist also zu erwarten, dass Polynome in monomialer Basis deutlich schwieriger zu beherrschen sind als Legendre-Polynome, wodurch der kumulative Vorhersagefehler bei Legendre-Polynomen kleiner sein sollte.

B-Splines

Wie die Lagrange-Polynome nutzen auch Splines [8] eine Menge an Punkten zur Definition des Approximators. Über diesen Punkten werden stückweise Polynomzüge definiert, um die guten Interpolationseigenschaften von Polynomen niedrigen Grades nutzen zu können ohne dadurch die gesamte Ausdrucksstärke des Approximators zu beschränken. Die Elemente der Basis des Raum der stückweise definierten Polynome werden üblicherweise als Basis-Splines oder kurz B-Splines [8] bezeichnet wird. Durch die Trennung von Polynomgrad und Ausdrucksstärke sind für den Designer zwei Designentscheidungen zu treffen. Zum Einen ist der Grad des B-Splines zu bestimmen, zum Anderen sind die Punkte zu wählen über welche die B-Splines definiert werden sollen. B-Splines 0. und 1. Grades sind typische lokale Approximatoren. Je höher der Grad des B-Splines gewählt wird, desto globaler agiert der dadurch definierte Approximator.

Das Verhalten der Gram-Matrix A von B-Splines hängt stark von deren Grad ab. Für B-Splines 0. Grades bilden die Basisfunktionen ϕ_i ein orthogonales System, was zu einer Diagonalstruktur der Gram-Matrix A führt. Für B-Splines 1. Grades zeigt die Gram-Matrix im eindimensionalen Fall eine Tridiagonalstruktur. Mit steigendem Grad der B-Splines wächst auch die Ordnung der Diagonalstruktur

der Gram-Matrix und wird somit der Einheitsmatrix immer unähnlicher und schlechter konditioniert, was deren Beherrschbarkeit verringert.

Takagi-Sugeno Fuzzy Systeme

Takagi-Sugeno Fuzzy Systeme (TSFS) [2] reihen sich in die Linie von Polynomen und B-Splines ein, da TSFS im Allgemeinen den Eingangsraum segmentieren und in lokale Modelle unterteilen, zwischen denen interpoliert wird. Die lokalen Modelle sind Polynome und üblicherweise vom Grad 1. Die Interpolation wird häufig von B-Splines 1. Ordnung vermittelt oder durch die Interpolation mit Gaußglocken. Beschränkt man sich bei der möglichen Interpolation auf B-Splines, wie wir es hier tun wollen, so bilden TSFS die Verallgemeinerung von Polynomen und B-Splines, in der zwischen lokalen polynomiellen Modellen polynomiell interpoliert wird. Durch den Grad der B-Splines lassen sich dabei Anschlussbedingungen für den Übergang zwischen den lokalen Modellen festlegen, wie Stetigkeit oder Differenzierbarkeit. Der Polynomgrad der lokalen Modelle legt die lokale Ausdrucksstärke fest. Als Verallgemeinerung von lokalen und globalen Approximatoren ist ein TSFS im Allgemeinen eine Mischform aus beiden und als regionaler Approximator zu betrachten, da die lokalen Modelle jeweils eine Teilregion des Eingangsraums abdecken.

Die Struktur der Gram-Matrix A eines TSFS wird maßgeblich durch die Wahl der B-Spline Interpolation und des Grades der lokalen Modelle beeinflusst. Die lokalen Modelle geben dabei jeweils eine Basisstruktur vor, die der Gram-Matrix des jeweiligen Polynoms entspricht. Die Gram-Matrix der Interpolation gibt die Blockstruktur vor nach welcher die Basisstruktur angeordnet und gewichtet wird, um die Gram-Matrix des vollständigen TSFS zu erzeugen. Die enge Verknüpfung von Interpolation und lokalen Modellen spiegelt sich somit genau in der Struktur der Gram-Matrix wider. Im Allgemeinen ergibt sich dabei eine schlecht konditionierte Gram-Matrix A , die der Einheitsmatrix nicht ähnlich ist.

Fourierreihen

Die Fourierreihenentwicklung [10] für trigonometrische Funktionen wurde ursprünglich für die Beschreibung und Analyse periodischer Funktionen entwickelt und lässt sich für die On-line Regression nutzen, indem man die Periodenlänge auf das Intervall ausdehnt auf dem die Approximation durchgeführt wird. Die LIP-Darstellung von Fourierreihen basiert auf Paaren von Sinus- und Kosinus-Termen und bildet einen globalen Approximator.

Die durch die Fourierreihe vorgegebenen Basisfunktionen ϕ_i bilden eine orthonormale Basis und somit ist ihre Gram-Matrix A die Identität. Als Approximator sollten sich Fourierreihen daher ähnlich verhalten wie Legendre-Polynome. Unterschiede im Approximationsverhalten können aus der unterschiedlichen Eignung der jeweiligen Funktionenräumen stammen die zu approximierende Funktion darstellen zu können.

Untersuchungen

Die Untersuchungen zum Approximationsverhalten der hier betrachteten Klassen von Approximatoren sollen helfen gewisse Eigenschaften der Approximatoren zu bewerten und die oben aufgestellten Erwartungshaltungen zu überprüfen. Diese Erwartungen beziehen sich auf das Verhalten der Approximatoren untereinander und auf deren Ergebnisse in Bezug auf ihre Konditionszahl. Beim Vergleich verschiedener Klassen von Approximatoren ist vor allem auf deren prinzipielle Vergleichbarkeit hinsichtlich ihrer Komplexität zu achten, daher werden für alle Untersuchungen Approximatoren mit gleicher Ausdrucksstärke bzw. Anzahl von Parametern verwendet. So ist auch für die beteiligten Lernverfahren der Parameterraum und das damit verbundene Optimierungsproblem gleich dimensioniert.

Vergleich der Klassen von Approximatoren

Um die Vergleichbarkeit der Untersuchungen zu unterschiedlichen Klassen von Approximatoren zu erhöhen werden nicht nur Approximatoren mit gleicher Ausdrucksstärke verwendet, es werden darüber hinaus auch unterschiedliche Zielfunktionen approximiert. Aus jeder Klasse von Approximatoren wird jeweils eine Zielfunktion gewählt, welche die Charakteristika der jeweiligen Klasse darstellt und sich möglichst stark von den Zielfunktionen aus den übrigen Klassen unterscheidet. Diese vier werden ergänzt um eine weitere artifizielle Zielfunktion, die in keiner der betrachteten Klassen von Funktionen liegt und somit von keinem Approximator exakt wiedergegeben werden kann. Die resultierenden fünf Zielfunktionen werden einheitlich über dem Intervall $[-10, 10]$ definiert und sind in folgender Liste beschrieben:⁸

- Polynom: Koeffizienten in monomialer Basis $(1, x, x^2, \dots, x^8)$:
(1, 1/10, -8/25, 0.0, 2/125, 0.0, -4/15625, 0, 1/781250)

⁸ Für den Vergleich der Klassen von Approximatoren beschränken wir uns der Einfachheit halber auf eindimensionale Zielfunktionen. Die Aussagen sind mit Blick auf das Verhalten der Gram-Matrix auf eine höhere Dimensionalität übertragbar.

- B-Spline: Neun über dem Intervall $[-10, 10]$ gleichmäßig verteilte Knotenpunkte x_i für B-Splines 1. Ordnung $\beta^1(x)$ mit den Gewichten: $(-1.7, 0.93, 0.22, -1.9, 1.0, 0.32, 0.34, -0.24, 0.0)$
- TSFS: ES werden drei lokale Modelle 2. Grades an den Positionen $(-7.5, 0, 7.5)$ definiert. Zwischen diesen lokalen Modellen wird linear interpoliert. Die Parameter der lokalen Modelle in monomialer Basis $(1, x, x^2)$ lauten wie folgt:

Modell I bei $x = -7.5$: $(-0.4, 0.1, 0.02)$

Modell II bei $x = \pm 0.0$: $(0.5, 0.1, -0.01)$

Modell III bei $x = +7.5$: $(-2, 0.2, 0.001)$

- Fourierreihe: Die Basis der Fourierreihe bilden neben dem Mittelwert die Grundschwingungen von Sinus und Kosinus über dem Intervall $[-10, 10]$ sowie deren erste drei Oberwellen:

$$\left(1, \sin\left(\frac{x\pi}{10}\right), \cos\left(\frac{x\pi}{10}\right), \sin\left(\frac{2x\pi}{10}\right), \cos\left(\frac{2x\pi}{10}\right), \dots, \sin\left(\frac{4x\pi}{10}\right), \cos\left(\frac{4x\pi}{10}\right)\right).$$

Die zugehörigen Koeffizienten für diese Basis sind:

$$(0.1, -0.7, -0.44, -0.12, 0.06, -0.08, 0.76, 0.04, 0.88)$$

- Artificielle Zielfunktion: Als Beispiel für eine Zielfunktion, die in keiner der obigen Klassen exakt repräsentierbar ist verwenden wir die Funktion $f(x)$ über dem Intervall $[-10, 10]$:

$$f(x) = |x - 3| \cdot \exp\left(-\left(\frac{x}{3}\right)^2\right) + \frac{1}{4} \cos(x) \quad (7)$$

Diese Funktion ist nicht-monoton und hat eine nicht-differenzierbare Stelle bei $x = 3$. Sie deckt somit verschiedene Charakteristika der betrachteten Klassen von Approximatoren ab.

Für die Durchführung der Untersuchungen wird der Ablauf verwendet wie er im Abschnitt zur On-line Regression beschrieben ist. Der Strom von Lerndaten wird nach den oben beschriebenen Zielfunktionen erzeugt und ist hier für jedes Experiment auf 150 Elemente begrenzt. Die Eingaben x_t werden anhand einer Gleichverteilung über dem Intervall $[-10, 10]$ zufällig gezogen. Als Approximatoren für die On-line Regression werden hier die folgenden Konfigurationen gewählt:

- Polynome

Aus der Klasse der Polynome werden zwei Approximatoren genutzt. Beide sind jeweils 8. Grades und unterscheiden sich nur hinsichtlich ihrer Basis, die einmal aus Monomen und einmal aus Legendre-Polynomen besteht.

- B-Splines

Für die B-Splines wird ein Approximator gewählt, der eine lineare Interpolation auf gleichmäßig verteilten Stützstellen auf dem Intervall $[-10, 10]$ durchführt. Dies entspricht einer Anordnung von B-Splines 1. Ordnung auf den gleichförmig verteilten Knotenpunkten. Die Positionen der Knotenpunkte entsprechen somit genau denen der Zielfunktion aus dieser Klasse, um deren exakte Darstellbarkeit zu ermöglichen.

- TSFS

Um für das TSFS die exakte Darstellbarkeit der Zielfunktion für diese Klasse zu erlauben werden drei lokale Modelle 2. Grades in monomialer Basis an den Positionen $(-7.5, 0, 7.5)$ platziert.

- Fourier

Der Approximator aus der Klasse der Fourier-Reihen verwendet die Oberwellen von Sinus und Kosinus bis einschließlich der 4. Ordnung über dem Intervall $[-10, 10]$. Seine Basis stimmt genau mit der in der Beschreibung der Zielfunktion definierten Basis überein.

Für jede Kombination aus Approximator und Zielfunktion werden die vier Lernverfahren PA-II, IRMA, RLS und SIRMA eingesetzt. Für jedes einzelne dieser Experimente, die durch Approximator, Zielfunktion und Lernverfahren eindeutig beschrieben sind, werden die Hyperparameter der Lernverfahren optimiert. Diese Optimierung wird auf den Mittelwerten des CSL über 1000 Wiederholungen eines dieser Experimente durchgeführt. Die hierzu verwendeten *random seeds* sind $r = 0, \dots, 999$. Folgende Hyperparameter der einzelnen Lernverfahren werden optimiert:

- PA-II: Der Hyperparameter $c \geq 0$ steuert die Aggressivität der Anpassungen des Lernverfahrens.
- IRMA: Die Steifheit der Ausgabe des Approximators wird über den Parameter $\sigma > 0$ geregelt und bestimmt wie stark das Wissen des Approximators bei einem Parameterupdate geschützt wird.

- RLS: Die Initialisierung der Autokorrelationsmatrix $S = \delta I$ des Lernverfahrens gibt das Maß der Ungewissheit an, die dem initialen Parametervektor des Approximators beigemessen wird.
- SIRMA: Zusätzlich zur (Grund-)Steifheit σ (aus IRMA) kommt bei SIRMA noch deren Veränderung τ über den Lernverlauf hinzu.

In allen Untersuchungen werden die Parameter der Approximatoren mit $\alpha = 0$ initialisiert. Die Performance der einzelnen Kombinationen aus Approximator, Zielfunktion und Lernverfahren wird normiert hinsichtlich der Performance der Vorinitialisierung, um Unterschiede in den Wertebereichen der Zielfunktionen auszugleichen und so die Vergleichbarkeit zu erhöhen. Ein Wert von eins entspricht somit der Performance ohne zu Lernen. Je kleiner der Wert ist, desto besser die Performance des lernfähigen Systems. Werte oberhalb von eins deuten auf Probleme beim Lernen hin.

Folgerungen aus dem Vergleich

Die Ergebnisse der Untersuchungen sind in Tabelle 1 zusammengefasst und nach Lernverfahren gruppiert. Innerhalb jeder Gruppe sind die besten Ergebnisse für die einzelnen Zielfunktion hervorgehoben⁹.

Die Folgerungen aus dem Vergleich werden zunächst auf allgemeine Eigenschaften der präsentierten Daten eingehen, anschließend werden die Erwartungshaltungen aus den Abschnitt zu Approximatoren überprüft und zum Abschluss werden einige bemerkenswerte Befunde für die verschiedenen Lernverfahren näher besprochen.

Aufgrund der Gestaltung der Untersuchungen hat jede Klasse von Approximatoren bei jeweils einer Zielfunktion den Vorteil diese exakt darstellen zu können. Es ist also zu erwarten, dass jeder Approximator für die Zielfunktion aus seiner Klasse das beste Ergebnis liefert. Diese Erwartung wird nur von den B-Splines bestätigt. Polynome und Fourierreihen gewinnen bei jeweils drei von vier Lernverfahren und TSFS sogar nur bei Zweien. Insgesamt zählen die B-Splines acht beste Approxiamtionen, die Fourierreihen sieben, Monomiale Polynome und TSFS jeweils zwei und Legendre-Polynome nur eine. B-Splines und Fourierreihen scheinen also für On-line Regression geeignete Approximatoren darzustellen.

⁹ Der extrem hohe Wert von 1426.532 des normierten CSL bei PA-II für die polynomielle Zielfunktion zeigt die Unverträglichkeit von PA-II und monomialen Polynomen, die hier sogar den Lernvorgang als Ganzes instabil macht.

Tabelle 1: Ergebnisse der Approximation verschiedener Zielfunktionen mit unterschiedlichen Approximatoren bei Verwendung der Lernverfahren PA-II, IRMA, RLS und SIRMA deren Hyperparameter für jedes Ergebnis einzeln optimiert sind. Die besten Ergebnisse hinsichtlich der Approximation einer bestimmten Zielfunktion sind für jedes Lernverfahren fett hervorgehoben.

Lernv.	Zielfkt.	Approximator				
		Monome	Legendre	B-Spline	TSFS	Fourier
PA-II	Polynom	1426.532	0.762	0.755	0.875	0.733
	B-Spline	0.943	0.375	0.085	0.991	0.261
	TSFS	0.990	0.143	0.066	0.793	0.119
	Fourier	0.814	0.486	0.113	0.942	0.065
	Artifiziell	0.951	0.347	0.178	0.960	0.067
IRMA	Polynom	0.703	0.763	0.745	0.778	0.732
	B-Spline	0.280	0.372	0.060	0.131	0.259
	TSFS	0.091	0.142	0.866	0.065	0.119
	Fourier	0.420	0.482	0.085	0.156	0.065
	Artifiziell	0.270	0.345	0.162	0.186	0.067
RLS	Polynom	0.979	0.697	0.722	0.863	0.704
	B-Spline	0.917	0.265	0.046	0.799	0.221
	TSFS	0.915	0.080	0.046	0.202	0.089
	Fourier	0.806	0.393	0.072	0.636	0.034
	Artifiziell	0.950	0.255	0.135	0.776	0.042
SIRMA	Polynom	0.694	0.713	0.747	0.805	1.472
	B-Spline	0.262	0.275	0.062	0.129	0.960
	TSFS	0.089	0.098	0.065	0.064	1.624
	Fourier	0.404	0.417	0.080	0.152	1.363
	Artifiziell	0.242	0.252	0.141	0.180	1.982
Kondition		$3.64 \cdot 10^{15}$	1.0	3.925	$8.42 \cdot 10^6$	1.0

Dieses Ungleichgewicht der Verteilung der besten Ergebnisse wird im Wesentlichen durch die Lernverfahren PA-II und RLS verursacht. Bei SIRMA zeigt sich eine deutliche Schwäche der Fourierreihen, die jedoch weniger dem Approximator als dem Lernverfahren anzulasten ist, da die von SIRMA durchgeführte Dichteschätzung mittelbar den unterliegenden Approximator verwendet und die periodischen Basisfunktionen der Fourierreihe eine exakte Lokalisierung der Lerndaten unmöglich machen¹⁰. Abgesehen von den Fourierreihen bei SIRMA erfüllen die Approximatoren bei IRMA und SIRMA die Erwartungen hinsichtlich der besten Ergebnisse. Die Unterschiede zwischen Monomialen und Legendre-Polynomen werden hier ignoriert und im Verlauf separat besprochen. Die Bevorzugung von B-Splines und Fourierreihen findet also gerade bei PA-II und RLS statt.

PA-II und RLS sind die beiden Lernverfahren, die mit der originalen Basis des Approximators arbeiten. IRMA und SIRMA führen über die Matrix A mittelbar eine Transformation der Basis des Approximators durch¹¹. Die transformierte Basis wird im Wesentlichen durch das Inverse der Gram-Matrix A erzeugt, sodass die Basis, auf der IRMA und SIRMA effektiv arbeiten, eine orthonormale Basis ist.

Die Ergebnisse der Untersuchungen zu den Lernverfahren PA-II und RLS erlauben es also die im Abschnitt zu den Approximatoren aufgestellten Erwartungen hinsichtlich des Einflusses der Konditionszahl der Gram Matrix eines Approximators zu überprüfen. Eine hohe Konditionszahl deutet auf eine wenig orthonormale Basis und somit auf einen schwer zu beherrschenden Approximator hin. Diese Erwartungen werden bei RLS voll erfüllt. Die Polynome in monomialer Basis, also hoher Konditionszahl, sind durchweg kaum besser als die Vorinitialisierung. Das beste Ergebnis für die Zielfunktion aus der Klasse der Polynome liefern die Legendre-Polynome, die eine orthonormale Basis bieten. Die Ergebnisse der anderen Approximatoren für diese Zielfunktion ordnen sich nach ihrer Konditionszahl. Die Fourierreihen schneiden hier fast so gut ab wie die Legendre-Polynome, sodass die Unterschiede auf die Approximierbarkeit der Zielfunktion zurückgeführt werden können. Das gleiche Verhalten hinsichtlich der Ordnung der Konditionszahlen und der Ordnung der Ergebnisse zeigt sich auch für die Zielfunktion aus der Klasse der B-Splines. Bei der TSFS-Zielfunktion stellt der B-Spline Approximator das beste Ergebnis, obwohl er keine orthonormale Basis bietet. Hier zeigt sich, dass

¹⁰ Die Dichteschätzung von SIRMA ist auf lokale oder wenigstens lokalisierbare Basisfunktionen angewiesen. Fourierreihen sind somit der denkbar schlechteste Approximator, um ihn mit SIRMA zu kombinieren.

¹¹ Die Unterschiede zwischen PA-II und IRMA lassen sich nicht durch einen Basiswechsel allein erklären. Dieser ist jedoch ein wesentlicher Bestandteil zur Erklärung der Unterschiede. Auf die weiteren Details wird an dieser Stelle nicht weiter eingegangen, da sie mehr über die Lernverfahren als über die Approximatoren aussagen.

neben der Konditionszahl die Lokalität der Basisfunktionen des Approximators von entscheidender Bedeutung für den Erfolg des On-line Lernens ist. Abgesehen von den B-Splines verhalten sich die Ergebnisse gemäß der Konditionszahlen der Approximatoren und der prinzipiellen Approximierbarkeit. Das ist auch bei der Zielfunktion aus der Klasse der Fourierreihen der Fall.

Für PA-II gelten im Wesentlichen die gleichen Aussagen wie für RLS, sodass die Bedeutung der Konditionszahl der Gram-Matrix eines Approximators und dessen Lokalität auch von diesen Ergebnissen gestützt wird. Die Ausnahme stellt das Ergebnis der Fourierreihe bei der Approximation der polynomiellen Zielfunktion dar. Eine mögliche Ursache für die Stärke der Fourierreihe hier und im Allgemeinen könnte die Art ihrer impliziten Extrapolation von Lerndaten sein. Die periodischen Basisfunktionen übertragen bestimmte lokale Eigenschaften wie Monotonie und deren Wechsel beim Lernen unmittelbar auf den gesamten Eingangsraum. So beeinflusst das Wissen eines einzelnen Lerndatums nicht nur einen Bereich um die Eingabe, sondern mehrere. Diese Art der Extrapolation von Wissen scheint in diesem konkreten Fall von Vorteil zu sein, kann jedoch auch von Nachteil sein, wenn die Art der Extrapolation des Approximators nicht zu den Eigenschaften der Zielfunktion passt.

Die Ergebnisse von IRMA und SIRMA lassen keine direkten Rückschlüsse auf die Bedeutung der Konditionszahl zu, da sie nicht auf dem originalen Approximator lernen. Da bei diesen Lernverfahren mittelbar auf Approximatoren in einer orthonormalen Basis gelernt wird, zeigt der Vergleich der verschiedenen Klassen von Approximatoren eher deren prinzipielle Approximierbarkeit bzw. allgemeine Ähnlichkeit. Umso interessanter ist der Vergleich der beiden polynomiellen Approximatoren bei diesen Lernverfahren. Während bei PA-II und RLS die Legendre-Polynome deutlich bessere Ergebnisse zeigen als die monomialen Polynome, ist diese Relation bei IRMA und SIRMA gerade umgekehrt. Das zeigt, dass es zwischen allen orthonormalen Basen immer noch Unterschiede hinsichtlich ihrer Eignung zur On-line Regression gibt und die Konditionszahl nur ein Kriterium zu deren Bewertung ist. Ein Unterschied zwischen der orthonormalisierten monomialen Basis und den Legendre-Polynomen ist der polynomielle Grad der einzelnen Basisfunktionen ϕ_i . Bei den Legendre-Polynomen hat jede Basisfunktion einen anderen Grad. Bei der mittelbar durch IRMA erzeugten Basis sind alle Basisfunktionen ϕ_i vom gleichen Grad, dem des Approximators. Dadurch können die orthonormalisierten Basisfunktionen genau die wie Lagrange-Polynome lokalisiert sein. Ob die Lokalität der Basisfunktionen die Eigenschaft ist, welche die Unterschiede erklärt oder Extrapolationseigenschaften eine Rolle spielen, wie sie bei den Fourierreihen besprochen wurden, kann im Rahmen dieser Arbeit nicht geklärt werden. Es könnten auch bisher nicht betrachtete Eigenschaften relevant sein.

Die Ergebnisse zur artifiziellen Zielfunktion unterstreichen die Dominanz der Fourierreihen und B-Splines und die Bedeutung von Kondition und Lokalität eines Approximators. Die Ordnung der Ergebnisse für PA-II und RLS sowie für IRMA und SIRMA entspricht derjenigen aus dem Vergleich der Klassen von Approximatoren untereinander und passt somit auch zur Ordnung der Konditionszahlen, die unten in der Tabelle aufgeführt sind. Der extrem hohe Wert der Konditionszahl von monomialen Polynomen von $3.64 \cdot 10^{15}$ spiegelt deren schlechte Ergebnisse für PA-II und RLS wider. Die Ergebnisse des TSFS insgesamt zeigen, dass der Approximator auch bei einer recht hohen Konditionszahl von $8.42 \cdot 10^6$ noch beherrschbar ist, jedoch nicht so einfach wie z.B. die B-Splines. Die Konditionszahl kann also durchaus hohe Werte annehmen, ohne dass die Beherrschbarkeit des Approximators sofort leidet.

Zusammenfassung und Fazit

Diese Arbeit führt die Konditionszahl der Gram Matrix A zur Beurteilung von verschiedenen Klassen von Approximatoren in Hinblick auf bestimmte Basen dieser Klasse und deren Verhalten ein. Die prognostizierte Bedeutung der Konditionszahl für den Erfolg des Lernens wurde in den Untersuchungen an unterschiedlichen Beispielen belegt. Darüber hinaus zeigen die Untersuchungen die Bedeutung der Lokalität der Basisfunktionen eines Approximators als ebenfalls relevant für den Erfolg des Lernens. Diese beiden Aspekte eines Approximators sind zwar nach den hier präsentierten Ergebnissen die wichtigsten, jedoch nicht die einzig relevanten. Die Extrapolationseigenschaften des Approximators sollten ebenso berücksichtigt werden und weitere hier nicht betrachtete Eigenschaften können ebenfalls eine Rolle spielen.

Außerdem hat sich gezeigt, dass Approximatoren mit einer Konditionszahl von eins, also einer orthonormalen Basis, zwar solide aber nicht notwendigerweise optimale Ergebnisse liefern. Das heißt, dass es weitere Eigenschaften wie zum Beispiel die Lokalität der Basisfunktionen gibt, welche die Eignung eines Approximators beeinflussen. Die Bedeutung der Kondition und Lokalität für typische Lernverfahren wie PA-II und RLS unterstreicht deren Relevanz für die Auswahl und Gestaltung eines Approximators für ein lernfähiges System. Lernverfahren wie IRMA und SIRMA, die mittelbar auf einer orthonormalisierten Basis arbeiten, integrieren diese Aspekte direkt in ihre Vorschrift zum Parameterupdate und können so helfen den Designprozess zu vereinfachen und robustere Ergebnisse zu erzielen.

Die Untersuchungen in Rahmen dieser Arbeit zeigen auch, dass die Wahl einer bestimmten Klassen eines Approximators nur dann einen signifikanten Einfluss auf den Erfolg des Lernens hat, wenn die Basis dieses Approximators eine niedrige Konditionszahl hat.

Literatur

- [1] Crammer, Koby, et al. „Online passive-aggressive algorithms.“ *The Journal of Machine Learning Research* 7 (2006): 551-585.
- [2] Buschermöhle, Andreas, and Werner Brockmann. „On-line learning with minimized change of the global mapping.“ *Evolving Systems* 6.2 (2014): 131-151.
- [3] Björck, Ake. *Numerical methods for least squares problems*. Siam, 1996.
- [4] Farrell, Jay A., and Marios M. Polycarpou. *Adaptive approximation based control: unifying neural, fuzzy and traditional adaptive approximation approaches*. Vol. 48. John Wiley & Sons, 2006.
- [5] Bronstein, Ilja N., et al. *Taschenbuch der Mathematik*. Eds. Eberhard Zeidler, and Wolfgang Hackbusch. Vol. 1. Springer-Verlag, 2012.
- [6] Hernández, M. A. „Chebyshev’s approximation algorithms and applications.“ *Computers & Mathematics with Applications* 41.3 (2001): 433-445.
- [7] Chan, W-Ch C., Ch-J. Chyan, and H. M. Srivastava. „The Lagrange polynomials in several variables.“ *Integral Transforms and Special Functions* 12.2 (2001): 139-148.
- [8] Prautzsch, Hartmut, Wolfgang Boehm, and Marco Paluszny. *Bézier and B-spline techniques*. Springer Science & Business Media, 2013.
- [9] Takagi, Tomohiro, and Michio Sugeno. „Fuzzy identification of systems and its applications to modeling and control.“ *Systems, Man and Cybernetics, IEEE Transactions on* 1 (1985): 116-132.
- [10] Seeley, Robert T. *An introduction to Fourier series and integrals*. Courier Corporation, 2014.

Relevanzbewertung technischer Informationen mittels Data-Mining Verfahren

Martyna Bator, Sahar Torkamani, Volker Lohweg

Institut für industrielle Informationstechnik

Hochschule Ostwestfalen-Lippe

Langenbruch 6, D-32657 Lemgo

E-Mail: {martyna.bator, sahar.torkamani, volker.lohweg}@hs-owl.de

Kurzfassung

Der vorliegende Beitrag beschäftigt sich mit einer neuartigen Methode zur effektiven Analyse und Bewertung von Patentdokumenten, die visuell durch eine interaktive Mensch-Maschine-Schnittstelle unterstützt wird. Es handelt sich um ein halbautomatisches Relevanzbewertung-Verfahren, das mittels eines umfangreichen Regelwerks relevante von nicht-relevanten Dokumenten unterscheiden soll, ohne die Intention des Anwenders kennen zu müssen. Unter Relevanz wird die Ähnlichkeit von Patentdokumenten in Bezug auf initial bewertet Referenzpatente verstanden. Dabei werden Text-Mining Techniken für die Datenvorverarbeitung, semantische und statistische Verfahren für die Merkmalbestimmung und ein Fuzzy-Pattern-Ansatz für die Patentbewertung herangezogen. Die einzelnen Verfahren sind so gewählt worden, dass sie im Hinblick auf eine spätere Endanwendung, den Anforderungen von Ressourceneffizienz und effektiver Implementierbarkeit genügen. Das Hauptaugenmerk des hier beschriebenen Regelwerks liegt darin, die Bedeutung der Patente verwertbar zu machen, automatisch zu ordnen und dem Anwender visuell verständlich darzustellen.

Einleitung

Patente schützen das geistige Eigentum von Erfindern und verhindern, dass ihre neuen Ideen kopiert werden. Sie sind von großer Bedeutung für den wirtschaftlichen Erfolg eines Unternehmens. Vor einer geplanten Patentanmeldung ist es wichtig festzustellen, ob eine bestimmte Technik bereits patentiert ist und wie die Erfolgsaussichten beurteilt werden können. Auch die Identifizierung von Verstößen

gegen eigene Patentanmeldungen ist für ein Unternehmen von äußerster Wichtigkeit. Aus diesem Grund wird immer mehr an Tools und Techniken geforscht und entwickelt, die einen Patentrechercheur bei der Suche unterstützen.

In den folgenden Kapiteln wird zunächst auf den Stand der Technik eingegangen. Darauf folgt die Erläuterung von den Text-Mining-Techniken: *Tokenisierung*, *Stoppwort-Eliminierung*, *Wortreduktion* [1], die für die Datenvorverarbeitung verwendet wurden, um eine Vergleichbarkeit der Datensätze sicherzustellen. Anschließend erfolgt eine Beschreibung der statistischen und semantischen Verfahren, die für die Extraktion von Merkmalen verwendet wurden. Im darauf folgenden Kapitel wird erläutert, wie die abschließende Bewertung von Patentähnlichkeiten mittels eines Fuzzy-Pattern Ansatzes [2] realisiert wurde. Dieser vermeidet Wahr- oder Falschaussagen, indem er die Bestimmung einer graduellen Zugehörigkeit ermöglicht. Damit eignet er sich als Relevanzmaß. Anschließend wird die Mensch-Maschine-Schnittstelle vorgestellt, die dem Anwender das Ergebnis der Patentbewertung visuell dargestellt. Im darauf folgenden Kapitel werden die Ergebnisse vorgestellt und abgeschlossen wird mit einer kurzen Zusammenfassung und Ausblick.

Stand der Technik

Verschiedene Techniken und Tools sind entwickelt worden, um Patentanalyse-Experten, Managern und Patent- und Markenämter bei den unterschiedlichsten Anforderungen im Bezug auf eine Patentrecherche zu unterstützen. Schlagwortbasierte Methoden [3, 4, 5] benötigen vordefiniert Schlagworte, um Patente zu finden und zu analysieren. Das Problem dabei ist, das Experten und ein spezielles Lexikon für die Definition der Schlagworte benötigt werden. Dabei konzentrieren sich diese Methoden auf einen Einsatzbereich, wie z.B. Medizin oder Pharmazie. Eine weitere Technik zur Patentanalyse liefern regelbasierte Verfahren [6]. Diese Methoden verwenden häufig Wenn-Dann-Regeln, um geeignete Informationen aus den Patenten zu filtern. Problematisch sind diese Methoden für große Datensätze, da die Anzahl der benötigten Regeln steigt und die Gefahr von Regelüberschreibung besteht. Außerdem benötigen viele Regeln, viel Speicherkapazität was im Widerspruch zu Ressourceneffizienz steht. Außerdem können diese Methoden mit unvollständigem Wissen nicht umgehen. Semantik-basierende Verfahren [7, 8] sind abhängig von Fachwissen. Diese Methoden haben Schwierigkeiten mit der Struktur von natürlicher Sprache. Aus diesem Grund können solche Verfahren die Konzepte der Patente nicht genau wiedergeben. Außerdem beziehen sich diese Verfahren auf Fachwissen aus einem Bereich und können nur schwierig auf andere Bereiche übertragen werden.

Die eben genannten Schwierigkeiten der einzelnen Verfahren, wie Speicherkapazität, Domain-basiertes Wissen und Zeiteffizienz werden mit dem hier vorgeschlagenen Verfahren behoben. Der Unterschied zu vorhandenen Verfahren besteht darin, dass die vorhandene Datenmenge nicht in Cluster unterteilt [9] wird, sondern dass ein Muster in der kognitiven Bewertung, die zunächst nicht-automatisiert erfolgt, erkannt wird. Die Idee hierbei ist es, die anfängliche manuelle Sichtung für das spätere Ranking zu nutzen. Auf diesem Gebiet gibt es Ansätze, um die große Anzahl an Patentschriften semantisch analysieren zu können [10, 11, 12, 13]. Dabei sind Rechercheure solchen "Black Box"-Modellen eher skeptisch gegenüber eingestellt, da das automatisierte Bewerten von komplexen Textinhalten zu einer fehlerhaften Einschätzung führen kann.

Ein weiterer Unterschied ist, dass diese in der Regel mit eigenen Datenbanken arbeiten oder sich auf einen Einsatzbereich fokussieren. Im Unterschied zu anderen Patentranking-Verfahren wie z.B. PatentMiner, welches mit strukturierten Daten, wie Erfinder, Anmelder und Zitationsinformationen [14] verwendet, arbeitet die hier vorgestellte Methode mit digitalisierten Informationen, die in einer nicht formalisierten Struktur vorliegen. Aus diesem Grund können Computerprogrammen nicht über eine einzelne Schnittstelle aggregiert auf diese Daten zugreifen.

Verfahren

In diesem Kapitel werden die einzelnen Schritte des Verfahrens vorgestellt. Zunächst werden einige Erläuterungen zu Patentschriften und der Datenakquise vorgenommen.

Patentschriften und Datenakquise

Jede Offenlegungs- oder Patentschrift hat dieselbe Struktur. Neben dem Titelblatt mit bibliografischen Angaben, wie z.B. *Inhaber*, *Anmelde-* und *Erteilungsdatum*, besteht sie aus einer *Zusammenfassung*, den *Patentansprüchen*, einer *Beschreibung* und *Zeichnungen*.

Die Patentansprüche definieren den Schutzzumfang und stellen somit das wichtigste Element des Patents dar. Sie definieren den Gegenstand, für den der Patentinhaber ein Verbotungsrecht gegenüber Dritten hat. Die Patentansprüche gliedern sich in unabhängige und abhängige Ansprüche. Erstere definieren den konkrete Schutzzumfang, welcher sich auf die maßgeblichen Eigenschaften der Erfindung bezieht. Zweitere setzt die Verfeinerungen der Erfindung fest. Patentansprüche

definieren eine Kombination von abstrakten Merkmalen und der Schutzbereich umfasst alle Produkte, die diese Merkmale beinhalten. Die Patentbeschreibung und die Zeichnungen dienen lediglich der Veranschaulichung und beschreiben die Begrifflichkeiten in den Ansprüchen eines Patents. Der Titel und die kurze Zusammenfassung bieten einen ersten Überblick um den grundsätzlichen Inhalt des Patents. Ob die dort gezeigten Aspekte allerdings auch vom Schutzzumfang des Patents umfasst sind, lässt sich mit diesen Informationen allerdings nicht beurteilen.

Die Basis für die Textdatenerfassung stellen englischsprachige Patentschriften dar, die mittels einer Patentsuchmaschine, die öffentlich zugängliche Patentdatenbanken (depatisnet [15], espacenet [16], uspto [17], google patents [18]) abfragt, ermittelt. Dabei wurde aus einer booleschen Suchanfrage eine unsortierte Kollektion \mathcal{D} an Patentschriften generiert.

Die hier vorgestellte Ansatz berücksichtigt die bibliografischen Daten: Titel, kurze Zusammenfassung und unabhängigen Ansprüche. Es ist naheliegend, das Informationen über den grundsätzlichen Inhalt eines Patents kombiniert mit den maßgeblichen Merkmalen der Erfindung, das größte Potential für eine effektive Relevanzbewertung liefern. Zusätzliche patentspezifische Daten werden aus der textuelle Information, die sich hinter dem Klassifikationssymbol der internationalen Patentklassifikation (IPC) verbirgt. Diese wurden aus einer externen Datenbank der World Intellectual Property Organization (WIPO) [19] extrahiert. Diese Daten sind kein direkter Bestandteil einer Patentschrift, verfügen jedoch über Informationen, die die Bewertung der Patentschriften in Bezug auf Ähnlichkeit unterstützen.

Datenvorverarbeitung

Um die Daten zu strukturieren und damit eine Vergleichbarkeit sicherzustellen, werden die Patentdokumente, die im XML-Format vorliegen, zunächst in ein für die Weiterverarbeitung brauchbares Format überführt. Dies geschieht mit Hilfe eines Parsers, der die relevanten bibliografischen Daten herausfiltert. Das Ergebnis ist eine Struktur von $\Psi = \{T, A, Z, Q\}$, in der jedem Patent x seine bibliografische Daten Titel T , unabhängige Ansprüche A , Zusammenfassung Z und IPC-Beschreibung Q eindeutig in Form von Textsequenzen zugeordnet sind

$$x := \{x^T, x^A, x^Z, x^Q\} \in \mathcal{D} \quad (1)$$

mit \mathcal{D} den Patentdokumenten.

Tokenisierung und Stoppwörter

Nach der Extraktion werden die Textsequenzen zerlegt. Alle Satzzeichen, Tabulatorabstände und Absätze werden als Trennzeichen benutzt. Die Textsequenzen werden tokenisiert, d.h. in Tupel zerlegt. Ein n -Tupel ist eine Zusammenfassung von n aufeinanderfolgenden Wörtern w_1, \dots, w_V einer Textsequenz eines Patents x , wobei die Reihenfolge von Bedeutung ist.

$$x := \begin{cases} x^T = (w_j^T) \\ x^A = (w_j^A) \\ x^Z = (w_j^Z) \\ x^Q = (w_j^C) \end{cases}, j = 1, \dots, N \quad (2)$$

Ein 1-Worttupel w_j ist die elementarste Einheit einer Patentschrift d . Zusätzlich werden die Textsequenzen in 2-Worttupel ${}^2w_e := (w_j, w_{j+1})$ und 3-Worttupel ${}^3w_l := (w_j, w_{j+1}, w_{j+2})$ mit $e = 1, \dots, E, l = 1, \dots, L, L < E < N$ zerlegt.

1-Worttupel, die häufig vorkommen, jedoch wenig Aussagekraft besitzen, können als Rauschen interpretiert werden [1]. Diesem Problem wird durch den Einsatz von Stoppwortlisten entgegengewirkt. Worttupel $\tau = \{w_j, {}^2w_e, {}^3w_l\}$, die Bestandteile dieser Liste aufweisen, werden entfernt. Stoppwörter haben keine Aussagekraft über den Textinhalt und können ohne Informationsverlust entfernt werden und zugleich wird die Anzahl der Daten reduziert.

Wortreduktion

Um die Anzahl der zu berücksichtigenden Patente zu erhöhen, sowie die Anzahl der Worttupel ohne wesentlichen Bedeutungsverlust zu reduzieren, wird ein Verfahren zur Bildung von Wortstämmen verwendet. Tupel, die sich nur durch einen anderen Pre- bzw. Suffix unterscheiden, werden zu einem Tupel zusammengefasst. Dabei ist nicht nur die Grundform eines Wortes gemeint, sondern auch nur ein um den Suffix und/oder Prefix eines Wortes gekürzten Wortstumpf. Der Wortstamm eines Wortes präsentiert grundsätzlich die Bedeutung eines Worttupels und semantische Abwandlungen, wie z.B. durch Pluralbildung oder Deklination, keine Abweichung vom eigentlichen Sinn bedeuten. Aufgrund der weitgehenden Sprachneutralität wird der Dice-Koeffizient [20] verwendet. Dieser gibt an, wie ähnlich sich zwei Worttupel sind, indem der Anteil der Trigramme ermittelt wird, die beide Worttupel enthalten. Ein Trigramm ist eine Folge von *drei*

aufeinanderfolgende Buchstaben eines Worttupels. Für zwei Tupel aus τ , z.B. w_j mit $j = 1, 2$ liegt der Dice-Koeffizient

$$dice(w_1, w_2) = \frac{2|W(w_1) \cap W(w_2)|}{|W(w_1)| + |W(w_2)|} \quad (3)$$

immer zwischen 0 und 1. $W(w_j)$ ist die Menge der Trigramme des Tupels w_j . Je näher dieser Wert bei 1 liegt, umso ähnlich sind sich w_1 und w_2 . Ist dies der Fall, werden sie zu w_1 zusammengefasst. Dieses Vorgehen erfordert jedoch, dass die Tupel in τ alphabetisch sortiert sind.

Merkmalextraktion

Ist die Vorverarbeitung abgeschlossen, erfolgt die Merkmalextraktion. Die Basis für die Berechnung der Merkmale sind Patentschriften der initialen Bewertung. Die Menge der relevant bewerteten Patente wird mit h^+ , die Menge der nicht relevanten Patente mit h^- bezeichnet.

Bei dem statistischen Verfahren werden basierend auf h^+ für alle Textelemente der Menge Ψ und alle Tupel der Menge τ das Vorkommen $v_{h^+} = v(\tau^\Psi(h^+))$ mit der Gewichtung

$$\mathbf{m}(\tau^\Psi(h^+)) = \frac{v(\tau^\Psi(h^+))}{|v(\tau^\Psi(h^+))|} \quad (4)$$

berechnet. Analog erfolgt die Berechnung für h^- . Daraus ergeben sich $2 \cdot (|\Psi| + |\tau|)$ statistische Merkmale.

Dem semantischen Verfahren zur Merkmalextraktion zugrunde liegt das generative, probabilistische Wahrscheinlichkeitsmodell Latente-Dirichlet-Allokation (LDA) [21]. In der Anwendung ist es das meist verwendete Verfahren zur Themenmodellierung.

LDA geht von der Annahme aus, dass Dokumente aus verschiedenen Themen entstehen, wobei ein Thema definiert ist als Verteilung über einem festen Vokabular an Wörtern. Eine Menge an Dokumenten wird assoziiert mit K Themen, wobei jedes Dokument diese K Themen zu verschiedenen Anteilen aufweist. Da in der Realität nur die Dokumente bekannt sind, ist es das Ziel von LDA diese Themen aus den Daten zu lernen. Durch die Formulierung eines *Hidden-Variable* Modells wird dieses Ziel *angegangen*. Bei *Hidden-Variable* Modellen handelt es sich um strukturierte Verteilungen, bei denen beobachtbare Variablen mit verborgenen Zufallsvariablen interagieren. Dies geschieht in der Form, dass eine verborgene Struktur in den beobachtbaren Daten aufgestellt und anschließend mit probabilistischer a-posteriori-Inferenz gelernt wird.

Die beobachtbaren Variablen sind die Wörter jedes Dokuments. Die verborgenen Variablen sind die Themenstruktur, die Themen selbst sowie das Verhältnis dieser in den einzelnen Dokumenten. Die a-aposteriori-Verteilung der verborgenen Variablen gegeben der Dokumente bestimmt eine latente Themenzerlegung der Dokumentenmenge. Die Interaktion zwischen den Dokumenten und der latenten Themenstruktur äußert sich in dem probabilistischen generativen Prozess, den LDA annimmt.

Sei K die Anzahl der Themen, V die Gesamtzahl aller Wörter in \mathcal{D} , α ein positiver K -Vektor und λ ein Skalar. $\text{Dir}_V(\alpha)$ ist eine V -dimensionale Dirichletverteilung mit dem Parameter α und $\text{Dir}_K(\lambda)$ eine K -dimensionale symmetrische Dirichletverteilung mit dem Skalar λ , dann definiert dieser Prozess, wie Dokumente generiert werden:

1. Für jedes Thema:
 - a) Wähle eine Verteilung über den Wörtern $\beta_k \sim \text{Dir}_V(\lambda)$
2. Für jedes Dokument:
 - a) Wähle einen Vektor mit den Themenproportionen $\theta_x \sim \text{Dir}_V(\alpha)$
 - b) Für jedes Wort:
 - i. Wähle eine Themenzuordnung
 $z_{x,n} \sim \text{Multinomial}(\theta_x), z_{x,n} \in \{1, \dots, K\}$
 - ii. Wähle ein Wort $w_{x,n} \sim \text{Multinomial}(\beta_{z_{x,n}}), w_{x,n} \in \{1, \dots, V\}$

Die Themen $\beta_{1:K}$, das Themenverhältnis pro Dokument $\theta_{1:\mathcal{D}}$ sowie die Themenzuweisungen pro Wort $z_{1:\mathcal{D}}$ stellen die latente Themenstruktur der Dokumente dar. LDA liefert eine multivariate Verteilung über den beobachtbaren und latenten Zufallsvariablen. Die latente Themenzerlegung einer Dokumentsammlung entsteht aus der korrespondierenden a-posteriori-Verteilung der verborgenen Variablen gegeben der Dokumente $\mathbf{w}_{1:\mathcal{D}}$

$$p(\beta_{1:K}, \theta_{1:\mathcal{D}}, z_{1:\mathcal{D}}, \mathbf{w}_{1:\mathcal{D}}) = \frac{p(\beta_{1:K}, \theta_{1:\mathcal{D}}, z_{1:\mathcal{D}}, \mathbf{w}_{1:\mathcal{D}})}{p(\mathbf{w}_{1:\mathcal{D}})} \quad (5)$$

Gleichung (5) entspricht der Umkehrung des generativen Prozesses. Gegeben der vorhandenen Anzahl an Dokumenten, entspricht (5) der Verteilung der latenten Variablen, welche die Dokumente generiert haben. Der Zähler ist die multivariate Verteilung aller Zufallsvariablen. Dies kann leicht für jede Einstellung der verborgenen Variablen berechnet werden. Der Nenner ist die Randverteilung der Beobachtungen, d.h. die Wahrscheinlichkeit die Dokumentkollektion unter jeder

möglichen Themenstruktur zu sehen. Theoretisch wäre es möglich, durch die Aufsummierung der multivariaten Verteilung über jede mögliche Instanziierung der latenten Themenstruktur, also die Summe über alle Möglichkeiten ein Wort der Dokumentkollektion einem der K Themen zuzuordnen. Das entspricht einer Komplexität von $\mathcal{O}(e^{\mathcal{D}})$ und ist somit nicht effizient berechenbar. Aus diesem Grund wird der Stichproben-basierende Approximierungsalgorithmus *Gibbs-Sampling* [22] verwendet. Das Grundprinzip besteht darin, eine Markov-Kette [23] zu konstruieren, dessen stationäre Verteilung die gesuchte A-posteriori-Verteilung ist.

Die a-posteriori Themen werden durch ihre Term-Wahrscheinlichkeit pro Thema β dargestellt. Ein Thema wird visualisiert, indem die Terme nach ihren Wahrscheinlichkeiten sortiert werden. Die a-posteriori Themenverhältnisse $\theta_{x,k}$ und Themenzuordnungen $z_{x,n,k}$ werden verwendet, um die Themenzerlegung eines Dokuments darzustellen. LDA wird auf h^+ und h^- angewendet. Dabei wird das Thema mit der höchsten Wahrscheinlichkeit mit den ξ wahrscheinlichsten Wörtern w_j gewählt. Analog wird LDA auch auf die Tupel 2w_e bzw. 3w_l angewendet.

Algorithmus 1.: Gewichtvektor für Merkmale

Voraussetzung:

- 1: $\mathbf{v}_{h^+} = (\mathbf{v}_{1,h^+}, \mathbf{v}_{2,h^+}, \dots, \mathbf{v}_{M,h^+})$: Merkmal Vektor von h^+ ,
- 2: $\mathbf{m}_{h^+} = (\mathbf{m}_{1,h^+}, \mathbf{m}_{2,h^+}, \dots, \mathbf{m}_{M,h^+})$: Gewichtsvektor von \mathbf{v}_{h^+} ,
- 3: $\mathbf{v}_{x_i} = (\mathbf{v}_{1,x_i}, \mathbf{v}_{2,x_i}, \dots, \mathbf{v}_{M,x_i})$: Merkmal Vektor von x_i

```

4: function WEIGHTCALCULATOR( $\mathbf{v}_{h^+}, \mathbf{m}_{h^+}, \mathbf{v}_{x_i}$ )
5:   for  $c = 1$  until  $M$  do
6:     if  $\mathbf{v}_{c,x_i} \in \mathbf{v}_{c,h^+}$  then
7:        $\mathbf{m}_{c,x_i} = \mathbf{m}_{c,h^+}$ ;
8:     else
9:        $\mathbf{m}_{c,x_i} = 0$ ;
10:    Endif
11:     $\mathbf{m}_{c,x_i} = \sum \mathbf{m}_{c,x_i}$ ;
12:  Endfor
13:  Return  $\mathbf{m}_{x_i}$ ;

```

Um numerische Werte für die Anwendung eines Klassifikators zu erhalten, werden für jedes Dokument x_i der Dokumentkollektion \mathcal{D} die Merkmale nach Algorithmus 1 berechnet:

Klassifikation

Die Bewertung von Patentähnlichkeiten erfolgt mittels des modifizierten Fuzzy-Pattern-Klassifikators (MFPC) [24], wobei dessen Zugehörigkeitswerte als Relevanzmaß betrachtet werden. Zusätzlich ist der MFPC aufgrund seiner Robust- und Leistungsfähigkeit im Hinblick auf eine effektiven Implementierbarkeit sehr gut geeignet [2]. Für M verschiedene Merkmale lautet der Zugehörigkeitswert für ein Objekt x mit den Merkmalen $\mathbf{m}_d = \{m_1, \dots, m_M\}$ nach Aggregation mit dem geometrischen Mittelwert

$$\mu_{\text{MFPC}}(x) = \mu_{\text{MFPC}}(\mathbf{m}, \mathbf{p}) = 2^{-\frac{1}{M}} \sum_{i=1}^M d_i(m_i, \mathbf{p}_i) \quad (6)$$

mit

$$d_i(m_i, \mathbf{p}_i) = \left(\frac{|m_i - S_i|}{C_i} \right)^{D_i} \in [0, 1] \quad (7)$$

und

$$\mathbf{p}_i = (S_i, C_i, D_i), \mathbf{p} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_M], \quad (8)$$

definiert die Eigenschaften einer Zugehörigkeitsfunktion, wobei S der Schwerpunkt, C der Abstand der Randzugehörigkeit vom Schwerpunkt und D die Schärfe der Flanken der Potentialfunktion ist.

Angelernt wird der Klassifikator mit den Merkmalen der Patentdokumente, die die anfängliche manuelle Sichtung geliefert hat. Während dieser Lernphase wird der Parameter $C = (1 + 2p_{ce}) \cdot \Delta$ mit $p_{ce} \in [0, 1]$ und $\Delta = (\max_{i=1}^{\mathcal{D}} m_i - \min_{i=1}^{\mathcal{D}} m_i) / 2$ berechnet.

Für ein beliebiges Patent x_j aus der Patentkollektion \mathcal{D} ergibt sich der Zugehörigkeitswerte $\mu_{\text{MFPC}}(x_j)$ zur Klasse „relevant“ und $\tilde{\mu}_{\text{MFPC}}(x_j)$ zur Klasse „nicht relevant“. Das Bewertungsmaß wird definiert mit

$$B(x_j) = \frac{B_{0,x_j} - \min(B_{0,x_j})}{\max(B_{0,x_j}) - \min(B_{0,x_j})}, \quad (9)$$

wobei $B_{0,x_j} = \mu_{\text{MFPC}}(x_j) - \tilde{\mu}_{\text{MFPC}}(x_j)$. Allein der Zugehörigkeitswert $\mu_{\text{MFPC}}(x_j)$ als Rankingwert, eignet sich nicht, da es durchaus möglich ist, das mehrere Patente dieselben Zugehörigkeitswerte haben. Dies ist der Fall, wenn zu Beginn mehrere Patente als relevant bewertet wurden.

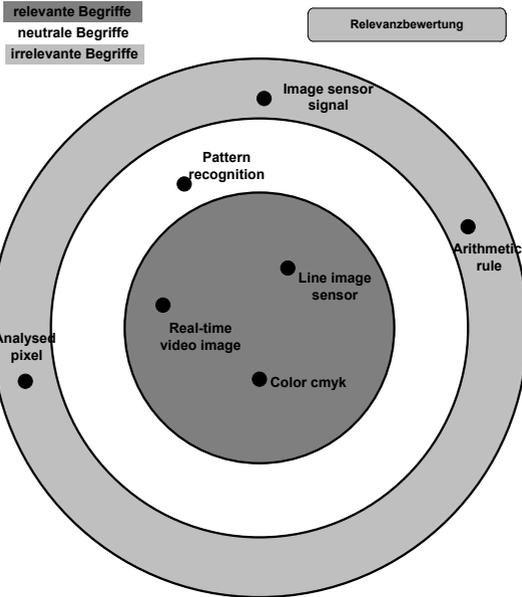
Das Ergebnis der Klassifizierung ist O_{Rank} , die Patentkollektion \mathcal{D} sortiert nach $B(x_j)$.

Visualisierung

Für die Visualisierung des Ranking-Ergebnisses D_{Rank} wurde eine grafische Mensch-Maschine-Schnittstelle (MMS) als Webapplikation entwickelt, welche in erster Instanz an den in Matlab vorliegenden Algorithmus angebunden ist. Der Vorteil einer solchen Applikation ist, dass sie sich auf jedem Computer, der über einen Browser verfügt, betreiben lässt, sodass die MMS als Plugin in eine Endanwendung eingebunden werden kann. Die MMS ermöglicht eine grafische iterative Verbesserung einer bereits durchgeführten Bewertung des hier beschriebenen Verfahrens.

Nach dem Start der Oberfläche werden die Ranking-Ergebnisse D_{Rank} der initialen Patentrecherche entgegengenommen. Im Anschluss werden die für das Ergebnis maßgeblichen Worttupel in einer Radaransicht dargestellt. Für die Darstellung werden die 10 relevantesten Patente $O_{\text{Rank}_{1:10}}$ und nicht relevantesten $O_{\text{Rank}_{\mathcal{D}-9:\mathcal{D}}}$ verwendet. Für $\tilde{h}^+ = \{O_{\text{Rank}_{1:10}}\}$ und $\tilde{h}^- = \{O_{\text{Rank}_{\mathcal{D}-9:\mathcal{D}}}\}$ werden, wie in dem Abschnitt Merkmalextraktion beschrieben, Worttupel τ berechnet, jedoch nur die auf statistischen Verfahren beruhen. Es soll dem Anwender angezeigt werden, welche Tupel besonders häufig in den neu gerankten Patenten vorkommen. Die Tupel, die am häufigsten in \tilde{h}^+ und \tilde{h}^- vorkommen, werden in den Kreisbahnen der Radaransicht dargestellt, dabei gilt $\tilde{h}^+(\tau) \neq \tilde{h}^-(\tau)$. In der mittleren Kreisbahn sind die Tupel dargestellt, die sowohl in \tilde{h}^+ als auch in \tilde{h}^- besonders häufig vorkommen. Die Häufigkeit des Vorkommens der dargestellten Tupel bezieht sich dabei auf die Dokumentfrequenz, die Anzahl der Dokumente in der Sammlung, die denen das Tupel enthalten ist. Die Radaransicht ordnet die Tupel auf drei konzentrischen Kreisringen an. Je größer der Abstand eines Tupels zum Zentrum ist, desto geringer ist seine Relevanz für das Ergebnis. Die äußere Kreisbahn beschreibt eine geringe Relevanz, die mittlere eine neutrale und das Zentrum eine hohe Relevanz in Bezug auf das Ergebnis. Zusätzlich wird die aus der Bewertung resultierende neu sortierte Patentliste angezeigt. Diese enthält die patentspezifischen Informationen der als am relevantesten eingestuften Patente in absteigender Reihenfolge. In der Qualität der Patentliste spiegelt sich der Erfolg des Rankingalgorithmus wieder. Plus- und Minusschaltflächen dienen der Bewertung der Zugehörigkeit des jeweiligen Patents in Bezug auf die Suchanfrage.

Legende:



Method for evaluating the quality of a printed matter, produced by a printing machine;

+ - CN100583147C

柯尼格及包尔公开股份有限公司

PROCESS PATENT – computing; calculating; counting

counting devices

cylinder presses

embossing decorations

- 1 The invention provides a method for evaluating the quality of a printed matter, produced by a printing machine, said printing machine producing several copies of the same printed matter, whereby an amount of copies is selected from the produced copies of said printed matter...

En authentication of security documents, in particular of banknotes

+ - AU2008256256A1

KBA GIORI SA

PROCESS PATENT – checking-devices

electrical signals

recognising patterns

handling record

- 2 There is described a method for checking the authenticity of security documents, in particular banknotes, wherein authentic security documents comprise security features; printed, applied or otherwise provided on the security documents, which security features comprise characteristic visual features intrinsic to the processes used for producing the security documents.

Bild 1: Schematische Darstellung der Mensch-Maschine-Schnittstelle

Um einen Informationsüberfluss in der Radardarstellung zu vermeiden, werden die Tupel, die sich aus der IPC-Beschreibung ergeben, in Form von Schaltflächen über der Patenzusammenfassung Z dargestellt (siehe Bild 1). Der Anwender kann seine Suche spezifizieren, indem er für ihn relevante Tupel in das Radarzentrum und nicht relevante außerhalb des Radars verschiebt. Innerhalb der Patentinformation gibt es zusätzlich eine Liste mit Schaltflächen, mit denen die Relevanz ausgewählter Tupel aus der IPC-Klassen-Beschreibung positiv oder negativ bewertet werden kann. Die Oberfläche der entwickelten MMS mit einem exemplarischen Rankingergebnis ist in Bild 1 dargestellt.

Mit Hilfe der MMS ist es möglich, die Bewertungsergebnisse zu begutachten, sowie durch einfache grafische Modifikationen der Bewertungsparameter, die Daten für einen weiteren Bewertungsdurchlauf aufzubereiten. Das bedeutet, die vom User ausgewählten Worttupel werden für eine erneute Bewertung den relevanten bzw. nicht relevanten Tupeln hinzugefügt. Ebenso werden die Mengen h^+ und h^- um die vom User nachträglich bewerteten Patente h_{User}^+ und h_{User}^- aus der neuen Patentliste erweitert, für einen neuen Bewertungsdurchlauf. Wenn $h_{\text{User}}^+ \in h^-$, dann $h^- = h^- \cap h_{\text{User}}^+$ und $h^+ = h^+ \cup h_{\text{User}}^+$. Ebenso wird verfahren, wenn $h_{\text{User}}^- \in h^+$ nur umgekehrt.

Dieser Vorgang kann beliebig oft wiederholt werden, um mit jeder Iteration ein verbessertes Rechercheergebnis zu erhalten. Auf eine erfolgreiche Neubewertung folgend, werden die neuen Ergebnisse der MMS wieder zur Verfügung gestellt. Dieser Vorgang kann sich beliebig oft wiederholen lassen, um mit jeder Iteration ein verbessertes Rechercheergebnis zu erhalten.

Ergebnisse

Anhand von 2 Patentdatensätzen, die das Ergebnis realer Recherchen aus unterschiedlichen Bereichen darstellen, wurde das hier beschriebene Verfahren auf seine Funktionalität und Güte getestet. Dabei wurden die Recherchen von professionellen Patent-Rechercheuren zunächst manuell gesichtet und teilweise in die Klassen relevant h^+ und nicht relevant h^- gruppiert. Das bedeutet, die Anzahl der Patente in h^+ bzw. h^- variiert.

Um dem Anwender die Arbeit zu erleichtern, bewertet er nur zwei Patente $|h^+| = 1$ und $|h^-| = 1$. Für den Algorithmus wird nur eine geringe Menge an Referenzpatenten als Grundlage für die Analysealgorithmen zu Verfügung gestellt. Für Datensatz 1, der insgesamt 56 Patente umfasst, ist das Ergebnis in Bild 2 dargestellt. Relevant bewertete Patente werden mit dem Symbol \square , nicht relevant bewertet mit dem Symbol \times dargestellt. In Bild 2 bezeichnet die

y -Achse den iterativen Workflow. Für $y(0)$ ist die zunächst unsortierte Patentliste mit den Positionen der bewerteten Patente dargestellt. Ein Patent ist umso relevanter, je niedriger seine Positionzahl in der Patentliste ist. Das bedeutet, im Idealfall befinden sich nach dem Relevanzranking alle relevanten Patente in dem linken Bereich, die nicht relevanten im rechten Bereich. An der Stelle $y(1)$ ist die Verteilung der Patente nach dem Relevanz-Ranking dargestellt. Bereits nach der ersten Stufe, ohne Mitwirkung durch den User, ist zu erkennen, dass sich die Klassen h^+ und h^- bis auf wenige Ausreißer gut separieren lassen. Das Gesamtranking jedoch ist für die geringe Anzahl an Referenzpatente bereits sehr gut. Dieses erste Ranking (Bild 2) wird dem User mittels der Web-Applikation dargestellt. Nun konnte er frei interagieren und seine Suche spezifizieren, in dem Wörter verschoben und Patente aus der neu sortierten Patentliste bewertet werden. Die erneute Relevanzbewertung ist in der iterativen Workflow(2) dargestellt. Der User hat zusätzlich ein relevantes und ein nicht relevantes Patent bewertet und einige Wörter in der Radaransicht verschoben. Die Klassen \square und \times lassen sich ohne Überschneidungen separieren.

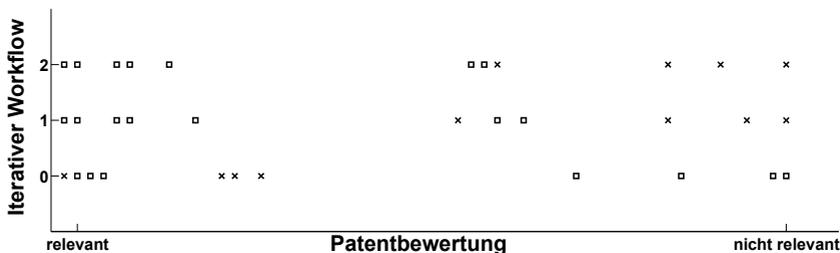


Bild 2: Datensatz 1: Ergebnisse mit 1 Referenzpatent pro Relevanzklasse



Bild 3: Datensatz 1: Ergebnisse mit 2 Referenzpatent pro Relevanzklasse

h^- durch den User bereits in einem Merkmal so stark unterscheiden, dass selbst ein Referenzpatent pro Klasse ausreichend ist.

Bei beiden Datensätzen konnte festgestellt werden, dass die Menge an Referenzpatenten h^+ und h^- die Treffergenauigkeit beeinflusst. Je mehr Referenzpatente vorhanden sind, desto höher die Treffergenauigkeit. Falls willkürlich bzw. irrtümlich Treffer als relevant markiert wurden, erhöhte sich in der ersten Iteration die Treffergenauigkeit nicht, da keine klaren Muster erkannt werden konnten. Im Zusammenspiel mit der Web-Applikation wird diese Fehlbewertung jedoch aufgehoben und in der zweiten Iteration ist das Ergebnis der Relevanz entsprechend bewertet. Für Datensatz 1 ist eine initiale Fehlbewertung in Bild 6 und für Datensatz 2 in Bild 7 dargestellt. Es ist deutlich erkennbar, dass in deinen Fällen die Bewertungsstufe 1 eine falsche Bewertung der Patente angibt. Die nicht relevanten Patente \times liegen weit in dem relevanten Bereich der Bewertung und die relevanten \square im nicht relevanten. Bewertet der Anwender in der zweiten Stufe mittels der MMS neu, erhält er ein Ranking, dass die für in relevanten Patente richtig bewertet. Der User muss nicht mehr die gesamte Liste an Patentdokumenten sichten, um für ihn relevante Patente zu erhalten, sondern erhält bereits nach wenigen Schritten eine Patentliste, die nach seinen Interessen sortiert ist.

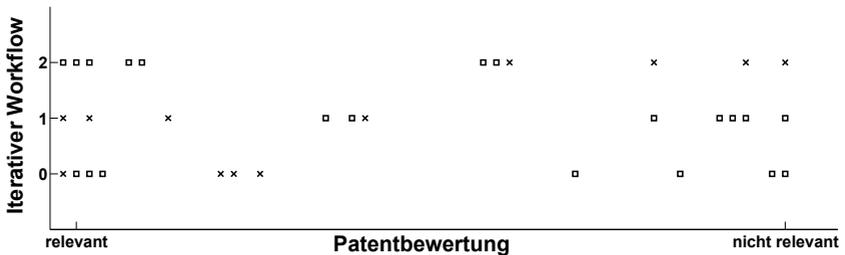


Bild 6: Datensatz 1: Ergebnisse mit einem Referenzpatent pro Relevanzklasse und initialer Falschbewertung durch den Rechercheur

Ein wesentlicher Aspekt bei der Realisierung waren die Anforderungen an Ressourceneffizienz und effektiver Implementierbarkeit. Für eine spätere Anwendung war es besonders wichtig, dass die Ausführung der Bewertung nicht zeitaufwendig ist. Das wiederum bedeutet, dass keine zeitaufwendigen Algorithmen verwendet werden sollten. Durch die Kombination des Bewertungsalgorithmus mit einer MMS wird in jedem Fall das initiale Ergebnis verbessert. Auch eine initiale Falschbewertung der Referenzpatente durch den Anwender, kann durch die Interaktion

mit der MMS abgefangen werden. Das wäre bei LDA, die ebenfalls Patente auf Ähnlichkeit untersuchen kann, nicht möglich. Prinzipiell wäre es möglich, die Themenverteilungsvektoren θ_{x_i} , die LDA liefert, zu nutzen, um die Patente mittels eines Abstandsmaßes auf Ähnlichkeit zu untersuchen. Eine fehlerhafte initiale Bewertung des Benutzers würde jedoch zu einem falschen Ranking-Ergebnis führen. Ferner ist mittels LDA die Untersuchung auf Ähnlichkeit nur mit einem Referenzpatent möglich. Enthält die initiale Bewertung des Recherchierenden mehrere Patente, müssten diese zu einem Themenvektor aggregiert werden. Wenn sich die Patente thematisch ähneln, sollte das prinzipiell möglich sein. Ist aber irrtümlich ein falsch bewertetes Patent in der Menge, kann keine korrekte Bewertung erfolgen. Aus den eben genannten Gründen lässt sich das hier vorgestellte Verfahren nicht wirklich mit anderen Verfahren vergleichen.

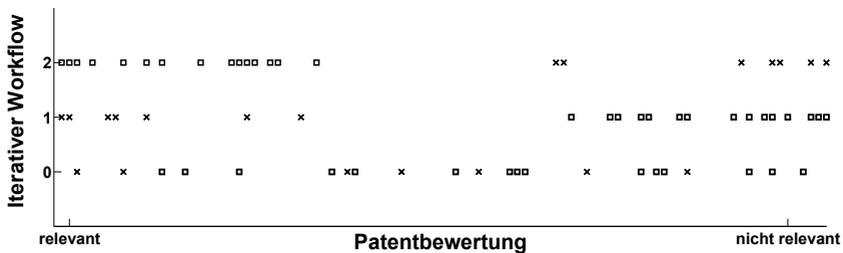


Bild 7: Datensatz 2: Ergebnisse mit einem Referenzpatent pro Relevanzklasse und initialer Falschbewertung durch den Recherheur

Zusammenfassung und Ausblick

In diesem Beitrag wurde ein Bewertungsverfahren entwickelt, das mit Unterstützung einer grafischen Webapplikation, einem Outlier bei der Suche nach für ihn relevanten Patentdokumenten unterstützt. Dabei wurden statistische und semantische Verfahren zur Bestimmung von Merkmalen verwendet, die dann mittels eines Fuzzy-Pattern-Ansatzes eine vorhandene Patentkollektion nach den Interessen des Rechercheurs bewertet. Der Benutzer hat eine Übersicht der wichtigsten Schlüsselbegriffe aus den für ihn relevanten Patenten aus der Trefferliste geclustert in die Bereiche relevant, neutral und irrelevant. Dieses wurde von den Nutzern als sehr positiv aufgenommen. Diese Begriffe könnten wiederum als Input für eine neue initiale Recherche verwendet werden, um die anfängliche Trefferliste noch spezieller für seine Recherche zu gestalten. Das bedeutet, dass

die anfängliche Trefferliste so spezifiziert wurde, dass sie nur für den Benutzer relevante Patente enthält.

Literatur

- [1] C. Aggerwal, C. Zhai: „Mining Text Data“. New York, Springer 2012.
- [2] V. Lohweg, C. Diederichs, D. Müller, Dietmar: „Algorithms for Hardware-Based Pattern Recognition“. In: *EURASIP Journal on Applied Signal Processing*, S.1912-1920. 2004
- [3] S. Liu, H. Liao, S. Pi, J. Hu: „Development of a patent retrieval and analysis platformea hybrid approach“. *Expert Systems with Applications* Vol. 38 Nr. 6. S. 7866-7868. Elsevier. 2011
- [4] J. Yoon, H. Park, K. Kim: „Identifying technological competition trends for R&D planning using dynamic patent maps: SAO-based content analysis“. *Scientometrics* Vol. 94. Nr. 1. S.313-331. Springer. 2013
- [5] S. Choi, H. Park, D. Kang, J. Lee, K. Kim: „An SAO-based text mining approach to building a technology tree for technology planning “. *Expert Systems with Applications* Vol. 39. Nr. 13. S. 11443-11455. Elsevier. 2012
- [6] M. Shih, D. Liu, M. Hsu: „ Discovering competitive intelligence by mining changes in patent trends “. *Expert Systems with Applications* Vol. 37. Nr. 4. S. 2882-2890. Elsevier. 2010
- [7] C. Lee, B. Song, Y. Park: „ How to assess patent infringement risks: a semantic patent claim analysis using dependency relationships“. *Technology Analysis & Strategic Management* Vol. 25. Nr.1. S.23-38. Taylor & Francis. 2013
- [8] W. Wang, C. Cheung: „A semantic-based intellectual property management system (SIPMS) for supporting patent analysis“. *Engineering Applications of Artificial Intelligence* Vol. 24. Nr. 8. S. 1510-1520. Elsevier. 2011
- [9] G. P. Michele Fattori: „Text mining applied to patent mapping: a practical business case“. In: *World Patent Information*. Vol. 23. Nr. 4. S. 335-342. Elsevier. 2003
- [10] J.-C. Lamirel, S. Al Shehabi, M. Hoffmann, C. Francois: „Intelligent patent analysis through the use of a neural network: experiment of multi-viewpoint analysis with the MultiSOM model“. In: *The ACL 2003 Workshop on Patent Corpus Processing*, S. 7-2. Sapporo, Japan. 2003

- [11] S. Sungjoo, B.Yoon, Y.Park: „An approach to discovering new technology opportunities: Keyword-based patent map approach“. In: *Technovation* 29, S.481-497 2009
- [12] Y.-H. Tseng, C.-J. Lin, Y.-I. Lin: „Text mining techniques for patent analysis“ In: *Information Processing & Management* 43, S.1216-1247. 2007
- [13] Y. Liang, R. Tan: „Text-Mining-based Patent Analysis in Product Innovative Process“ In: *Trends in Computer Aided Innovation - IFIP The International Federation for Information Processing* Vol. 250, S.89-96. Springer US, 2007
- [14] J. Tang, B. Wang, Y. Yang, P. Hu, Y. Zhao, X. Yan, B. Gao, M. Huang, P. Xu, W. Li, A. Usadi: „PatentMiner: topic-driven patent analysis and mining“. In: *Proceedings of the 18th ACM international conference on Knowledge discovery and data mining*, S.1366-1374. ACM, 2012
- [15] DPMA - Deutsche Patent- und Markenamt. <http://www.dpma.de/> Überprüfungsdatum 2015-09-23
- [16] Espacenet. <http://worldwide.espacenet.com/> Überprüfungsdatum 2015-09-24
- [17] uspto - United States Patent and Trademark Office. <http://www.uspto.gov/> Überprüfungsdatum 2015-09-24
- [18] Google Patents <https://www.google.de/> Überprüfungsdatum 2015-09-24
- [19] World Intellectual Property Organization <http://www.wipo.int/portal/en/index.html> Überprüfungsdatum 2015-09-23
- [20] M. M. Deza, E. Deza: „Encyclopedia of distances“. S. 326. Springer. 2009.
- [21] D. Blei, A. Ng, M. Jordan: „Latent Dirichlet Allocation“. In: *Journal of Machine Learning Research* 3, S. 993-1022. 2003.
- [22] M. Steyvers, T. Griffiths: „Latent Semantic Analysis: A Road to Meaning“. *Kap. Probabilistic topic models*. Lawrence Erlbaum. 2007.
- [23] E. Behrends: „Introduction to Markov Chains“. Vol. 228. Vieweg. 2000
- [24] V. Lohweg, U. Mönks: „Fuzzy-Pattern-Classifer Based Sensor Fusion for Machine Conditioning“. In: *Sensor Fusion and its Applications* (Thomas, C., Hg.), S. 319-346. InTech. 2010.

Depth Estimation in Monocular Images: Quantitative versus Qualitative Approaches

Ralph Ewerth¹, Alexander Balz², Jan Gehlhaar², Krzysztof Dembczyński³, Eyke Hüllermeier⁴

¹Jena University of Applied Sciences, ²University of Marburg, ³Poznań University of Technology, ⁴University of Paderborn
E-Mail: ralph.ewerth@fh-jena.de, {balz, gehlhaar}@informatik.uni-marburg.de, kdembczynski@cs.put.poznan.pl, eyke@upb.de

Introduction

Depth estimation in images and video sequences is an important prerequisite for a number of computer vision tasks and scene understanding. However, estimating the camera distance, i.e., depth information, of objects in monocular images is a difficult problem. Human visual perception of depth in single images relies on some monocular depth clues. Different approaches have been suggested to tackle the problem of automatic depth estimation in single images using regression, for example, by using conditional random fields (e.g. [8, 9]) or deep learning methods (e.g. [7]). Such approaches estimate the camera distance of scene parts and objects in a quantitative manner.

A natural question arising in this context concerns the suitability of the information contained in an image for training a quantitative regression model. More generally, one may wonder about the limits of an approach of that kind. Classical methods for three-dimensional image acquisition require additional information, for example obtained by a second camera or generated by structured light projection. There is a number of approaches of this kind, but they are not suitable for depth estimation in single images.

In this extended abstract, we discuss an alternative approach that is based on learning-to-rank methods from the field of preference learning [4]. The main idea is to qualitatively estimate the relative order of objects or image regions, respectively. This approach is motivated by the assumption that information extracted from a single image does not provide enough information about the correct (quantitative) camera distance. Moreover, what is typically needed in practice is indeed the relative order of scene objects, not the precise distances. Seen from this point of

view, a regression method is actually solving an unnecessarily difficult problem or, stated differently, tackles the actual problem only indirectly by solving a more difficult problem first. Finally, supervision becomes simpler for ranking methods, because qualitative comparisons are easier to obtain than numerical depth values, especially if humans are involved in the labelling of images.

In the presented approach, monocular depth features are needed as well. We suggest corresponding descriptors for some monocular depth criteria that are aimed at capturing information about the relative order of two regions in an image. These descriptors are then employed to compare and rank image regions using a ranking method. Rankboost is used in our first experiments. This approach is shown to be capable of qualitatively estimating depth in a single image. Based on these preliminary results, the advantages and disadvantages of a ranking-based approach are discussed with respect to several aspects: possibility of generating training data for different applications, space and runtime complexity, accuracy of estimation, and possible use cases.

Feature Extraction and Depth Estimation by Rankboost

In a first step, the image is partitioned into rectangular regions of equal size. Then, features are extracted for each region according to the modeled monocular clues. The following six monocular depth clues are used in the proposed system: relative height, atmospheric perspective, linear perspective, texture gradients (contours), and two kinds of color similarity.

Since the relative depth information is only weakly correlated with the monocular depth descriptors, an integration model is required that combines them in an optimal manner. As we tackle depth estimation as a learning-to-rank problem, a function is sought that orders image regions according to their camera distance, i.e. scene depth. Preference learning methods are designed for tasks of this kind [4]. More specifically, the problem of ranking image regions can be formalized as a so-called *subset ranking* or *object ranking* task, to which the Rankboost algorithm is applicable [5]. In general, boosting algorithms are meta learners that combine so-called weak learners in an ensemble. Each weak learner is expected to yield an error rate slightly better than random guessing. The decision of the ensemble is finally realized in the form of a linear combination of the weak learners.

Depth estimation in single images can now be tackled using Rankboost as follows. Each training image is considered as an exemplary object ranking, where the objects are image regions ranked according to their scene depth based on the given features. Each image region can be described via monocular depth features that

are used to infer a strong depth (ranking) function using Rankboost's training algorithm. The concept of combining weak rankings into a strong ranking function fits well to the depth estimation scenario, since the monocular depth clues itself provide only rather weak information about scene depth. A ranking of image regions based on a single monocular depth feature represents a weak ranking. The final depth map is learned by combining the best weak rankers, namely those with the minimum error in the corresponding training round.

Preliminary Experimental Results

We present some preliminary experimental results for depth estimation using a learning-to-rank approach. The publicly available data set provided by Saxena et al. [9] is used in these experiments. It consists of 534 outdoor images in total and is divided into a training set of 400 images and a test set of 134 images. The original image resolution is 1704×2272 , the resolution of the related ground truth depth maps is 305×55 . Each image is separated into image regions, and monocular depth descriptors are extracted for each region. The monocular depth clues are represented by a six-dimensional feature vector. This feature vector is used to generate a ranking-based depth map for an image that represents the ordering of image regions according to their scene depth.

We used the Rankboost implementation of Dang's Ranklib library [1]. In the Rankboost tests, 300 training rounds and 10 thresholds were used by default. The quality of generated depth maps is evaluated in terms of the error rate, which is the percentage of incorrectly ordered pairs of regions in a depth map representation of an image. The original resolution of depth maps has been scaled from 305×55 to 61×55 in order to have a more balanced ratio of width and height compared with the original images, for both the entire depth map and its regions.

The presented approach achieves an error rate of 23.4% on the test set. Some example results are shown in Figure 1. These results are promising and confirm that a qualitative estimation of relative depth in single images is in principle possible, although the accuracy of the best regression methods has not yet been reached. For example, we have evaluated Liu et al.'s approach [8], which achieves an error rate of 17.7%. It needs to be mentioned, however, that this approach uses additional information by predicting semantic categories for image regions to infer depth information.

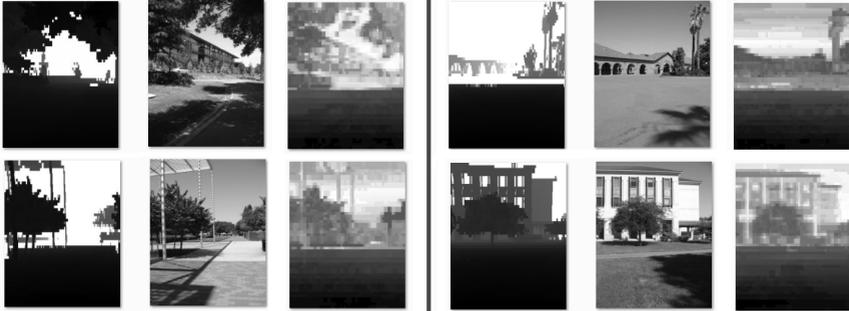


Figure 1: Some example results presented in two columns: ground truth depth map (left), original image (middle), and depth map created by Rankboost (right).

Conclusion: Comparing Regression and Ranking Methods

One of the main advantages of qualitative depth estimation is a broader range of application domains thanks to a simplification of data collection. For a qualitative approach, training data does not need to be acquired by a 3D-camera system but can easily be generated using manual labeling with an appropriate software tool. Hence, depth estimation becomes possible in domains for which no depth data has been generated before. One possible application domain lies in the field of visual concept detection in images or video retrieval [2].

In addition, the presented approach using Rankboost is efficient. Its run-time complexity is linear in the number of regions and training rounds, whereas, for example, solving the linear program for inference in Saxena et al.'s approach [9] has polynomial runtime complexity.

By now, the main disadvantage of the presented approach is that alternative methods still achieve a better accuracy. Hence, the main direction for future work is to model better monocular depth descriptors that capture depth information more accurately. Besides, further ranking methods ought to be investigated as alternatives to Rankboost.

References

- [1] V. Dang. “Ranklib – A Library for Ranking Algorithms (Version 1.1)”. <http://people.cs.umass.edu/~vdang/ranklib.html> 2010.
- [2] R. Ewerth, R., M. Schwalb, and B. Freisleben, B. “Using Depth Features to Retrieve Monocular Video Shots”. In: *Proceedings of the ACM International Conference on Image and Video Retrieval*. Amsterdam, The Netherlands. 210–217. 2007.
- [3] P.F. Felzenszwalb and D.P. Huttenlocher. “Efficient Graph-Based Image Segmentation”. In: *International Journal of Computer Vision*. Volume 59. Issue 2. pp. 167–181. Kluwer Academic Publishers. 2004.
- [4] J. Fürnkranz and E. Hüllermeier. “Preference Learning: An Introduction”. In: *Preference Learning* 4.1. pp. 1-17. Springer-Verlag. 2010.
- [5] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer. “An Efficient Boosting Algorithm for Combining Preferences”. In: *Journal of Machine Learning Research*. Vol. 4, December. pp. 933–969. MIT Press. 2003.
- [6] E. B. Goldstein. “Sensation and perception”. Wadsworth Cengage Learning. 2009.
- [7] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. “Depth and Surface Normal Estimation from Monocular Images Using Regression on Deep Features and Hierarchical CRFs”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Boston, USA. pp. 1119–1127. 2015.
- [8] B. Liu, S. Gould, and D. Koller. “Single Image Depth Estimation from Predicted Semantic Labels”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA. pp. 1253–1260. 2010.
- [9] A. Saxena, M. Sun, A.Y. Ng. “Make3D: Learning 3D Scene Structure from a Single Still Image”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 31, No. 5. pp. 824–840. 2009.

Convolutional Neural Networks für die semantische Segmentierung von Szenen

Malte Oeljeklaus, Frank Hoffmann, Torsten Bertram

Lehrstuhl für Regelungssystemtechnik, TU Dortmund
E-Mail: malte.oeljeklaus@tu-dortmund.de

Einleitung

Das Problem der semantischen Segmentierung beinhaltet die Unterteilung von Bildern in Regionen und deren Zuordnung zu Objektklassen. Verfahren unter Einsatz von Convolutional Neural Networks (CNN) gelten aktuell sowohl für diese Aufgabe, ebenso wie für ähnliche Probleme der Bildanalyse, als innovativer und in Benchmarks zur bildbasierten Objekterkennung besonders erfolgreicher Ansatz. Aus dem Einsatz von CNNs insbesondere für die semantische Segmentierung ergeben sich besondere Herausforderungen an ihr Training. So wird der Problematik der Überanpassung, im Zusammenhang mit der hohen Anzahl freier Parameter in CNNs, häufig durch besonders umfangreiche Trainingsdatensätze begegnet. Die Gewinnung einer großen Anzahl an Trainingsdaten ist für das semantische Segmentieren auf Grund des hohen Aufwands einer manuellen Markierung von Objektklassen wenig praktikabel. Weiterhin erfolgt in den zur Szenen- oder Objekterkennung eingesetzten gängigen Modellstrukturen in regelmäßigen Abständen eine Unterabtastung, wodurch die Modelle vereinfacht und der Speicherbedarf reduziert wird. Die semantische Segmentierung erfordert, wie in Abbildung 1 dargestellt, eine dichte Klassifikation der Bildpunkte auf Pixelebene. Solch eine hohe Auflösung steht jedoch im Widerspruch zu der schrittweisen Unterabtastung auf den verschiedenen Modellebenen. Der Beitrag liefert einen Überblick über die Grundlagen von CNN sowie über deren Anwendung auf das Problem der semantischen Segmentierung, welche den oben angesprochenen Herausforderungen Rechnung trägt. So zeigen aktuelle Arbeiten Strukturen auf, die zu effizienten und hoch genauen CNN Modellen mit vergleichsweise wenigen freien Parametern führen [1]. Das Problem der beschränkten Verfügbarkeit vollständig segmentierter und klassifizierter Bilddatensätze lässt sich durch eine Vorinitialisierung der Parameter mit Werten bereits auf anderen Bilddaten vortrainierter Modelle reduzieren [2]. Weiterhin lassen sich die Effekte der für die semantische Segmentierung störenden Unterabtastung durch Aufgabe einer streng hierarchischen Struktur

und die Einführung paralleler Netzwerkpfade umgehen [3]. Der Beitrag untersucht die skizzierten Methoden für das Problem der semantischen Segmentierung von Outdooransichten für die Umfelderkennung in Fahrerassistenzsystemen.

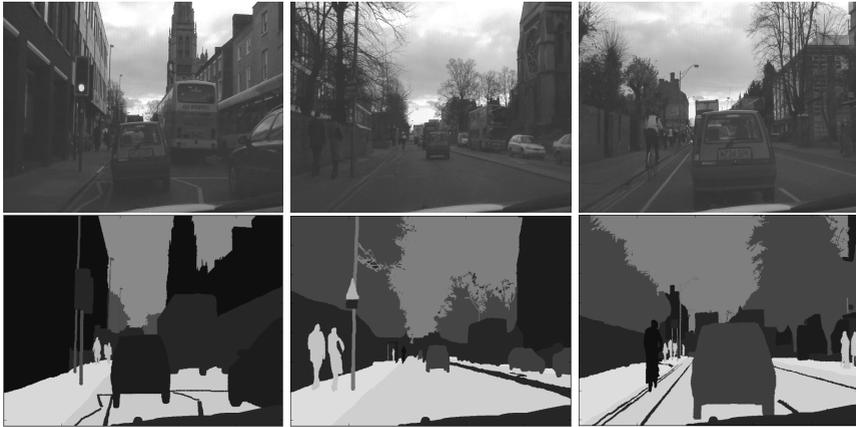


Bild 1: Problem der semantischen Segmentierung (Beispiele aus [10])

Verwandte Arbeiten

Dem Stand der Technik genügende Methoden zur semantischen Segmentierung basieren zumeist auf CNNs. [4] erzielen mit ihrer Methode eine auffallend genaue Segmentierung, welche die Genauigkeiten ähnlicher Ansätze wie [5] und [6] übertrifft. Die drei angesprochenen Methoden basieren auf der Kombination von CNNs und Conditional Random Fields (CRF). Die CNNs werden dabei für das Bestimmen einer Merkmalsrepräsentation eingesetzt, während die CRFs auf dieser Grundlage kontextuelle Zusammenhänge abbilden. Der Ansatz aus [6] verwendet eine Multiskalen-Pyramide, um Merkmalsrepräsentationen zu erhalten die jeweils nur in bestimmten Skalen auftreten. Die Arbeit in [7] verfolgt einen Ansatz, bei welchem die Klassifikation und die Segmentierung getrennt durchgeführt werden. Für jeden der beiden Schritte wird ein separates Modell trainiert, in dem zunächst ein skalarer Bezeichner der Klasse bestimmt wird und im Anschluss die Konturen des abgebildeten Objekts segmentiert werden. Durch das separate Training beider Teilmodelle können für das Klassifikationsmodell umfangreiche Datensätze verwendet werden, so dass das Problem der mangelnden Verfügbarkeit manuell

segmentierter Datensätze abgemildert wird. Aus der Beobachtung, dass die räumliche Auflösung in späteren Schichten der CNNs für das Problem der semantischen Segmentierung zu grob ist, entwickeln [8] einen Ansatz der als „Hypercolumn“ bezeichnet wird. Dabei werden Informationen aus allen Schichten des Netzwerks berücksichtigt, in dem die räumlich korrespondierenden Werte der Aktivierungsfunktionen des gesamten Netzwerk in einem Vektor zusammengefasst und im Sinne von Pixeldeskriptoren für die anschließende semantische Segmentierung verwendet werden. Dies unterscheidet den Ansatz zu der Multiskalen-Pyramide aus [6], in welcher die Aktivierungsfunktionen derselben Schicht auf mehreren Skalen berechnet werden. [3] adaptieren CNNs von einem reinen Klassifikationsproblem unter Beibehaltung der wesentlichen Modellparameter auf das Problem der semantischen Segmentierung und bezeichnen ihren Ansatz als „fully convolutional networks“. Der Ansatz kombiniert grobe semantische Informationen der späten Schichten mit hoch aufgelösten Konturinformationen der frühen Schichten. Hierzu wird eine Modellstruktur gebildet, in welcher ausgehend von frühen, fein aufgelösten Schichten parallele Netzwerkpfade ohne Unterabtastung und mit geringem zusätzlichem Rechenaufwand gebildet werden. Diese werden mit den semantischen Informationen der späten Schichten fusioniert, so dass schließlich die Ausgangswerte des Modells entstehen. Der wesentliche Vorteil des Ansatzes besteht in der Möglichkeit, die Parameter vortrainierter Modelle als Initialisierung beizubehalten und das Modell lediglich ausgehend von dieser Initialisierung an das konkret betrachtete Problem der semantischen Segmentierung anzupassen („fine-tuning“).

Convolutional Neural Network

In CNNs werden mehrere Verarbeitungsschichten in einer typischerweise hierarchischen Struktur hintereinander geschaltet, in jeder Schicht werden dabei die Ausgangsdaten der vorherigen Schicht weiterverarbeitet. Die einer Schicht n zugeordneten Daten werden dabei als dreidimensionale Matrix X_n der Größe $h \times w \times d$ vorgehalten, wobei h und w die zweidimensionalen Bildkoordinaten bilden und d die Bildkanäle bezeichnet, welche für die Eingangsdaten den Farbkanälen entsprechen. Einen besonderen Aspekt der Netzwerktopologie bildet das aus der Biologie motivierte Konzept der rezeptiven Felder (engl. „receptive fields“). Dieses besagt, dass zwischen den Schichten Verbindungspfade nur innerhalb einer lokalen Nachbarschaft existieren. Abbildung 2 veranschaulicht das Konzept der rezeptiven Felder. In den Verarbeitungsschichten lassen sich vielfältige mathematische Operationen realisieren. Die für diese Arbeit relevanten Typen von Verarbeitungsschichten sollen im folgenden kurz beschrieben werden.

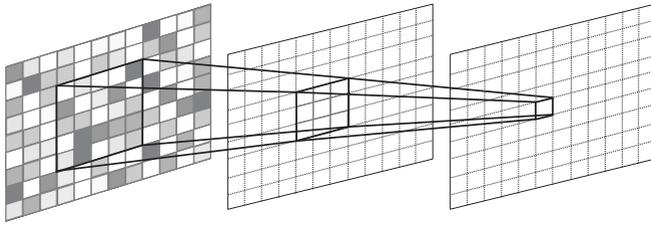


Bild 2: Konzept der rezeptiven Felder: Verbindungen zwischen den Schichten des neuronalen Netzes bleiben auf eine lokale Nachbarschaft beschränkt

Verarbeitungsschichten

Faltungsschicht: Die normalisierte, gewichtete Summe über die Bildpunkte des Eingangs eines rezeptiven Felds entspricht einer linearen Faltungsoperation, falls die Gewichtungsfaktoren für alle rezeptiven Felder identisch gewählt werden. Die Faltungsmatrix wird mit Θ_n bezeichnet, ihre Wiederverwendung für alle rezeptiven Felder führt auf eine deutlich reduzierte Anzahl freier Parameter, was die Parameteridentifikation hinsichtlich Speicherbedarf und erforderlicher Rechenzeit deutlich vereinfacht. Die Ausgangswerte dieser namensgebenden Faltungsschicht (engl. „convolutional layer“) ergeben sich somit aus einer gewöhnlichen diskreten, linearen Faltungsoperation:

$$\begin{aligned}
 X_n(h', w', d') &= X_{n-1} * \Theta_n(h', w', d') \\
 &= \sum_{i=1}^h \sum_{j=1}^w \sum_{k=1}^d X_{n-1}(i, j, k) \Theta(h' - i, w' - j, d' - k)
 \end{aligned}$$

Da die Eingangs- und Ausgangsdaten allgemein aus mehreren Bildkanälen bestehen, hat die Faltungsmatrix ebenfalls eine dreidimensionale Struktur. Im Allgemeinen werden die Filterantworten zu mehreren verschiedenen Faltungsmatrizen berechnet, so dass sich am Ausgang der Faltungsschicht wiederum mehrere Bildkanäle ergeben. Darüber hinaus lässt sich eine inverse Faltungsschicht (engl. „deconvolutional layer“) definieren, indem einzelne Bildpunkte mit einer Faltungsmatrix multipliziert und sämtliche entstehenden Werte in einem Ausgangsbild zusammengefasst werden. Diese Form der Aggregation wird in dieser Arbeit insbesondere dazu eingesetzt eine Überabtastung zu realisieren.

RELU Aktivierungsschicht: Im Kontext der CNN werden die nichtlinearen Aktivierungsfunktionen einer eigenen Schicht zugeordnet. Bei den traditionell

verwendeten Sigmoidfunktionen nimmt der Gradient für steigende Werte ab, was für eine effiziente Optimierung problematisch ist. Daher werden als Nichtlinearität sogenannte rectified linear units (RELU) eingesetzt:

$$X_n(h', w', d') = \max(0, X_{n-1}(h', w', d')) + \nu \cdot \min(0, X_{n-1}(h', w', d'))$$

Diese weisen für alle Werte von X_{n-1} einen signifikanten Gradienten auf, der Wert des Gradienten für negative X_{n-1} ergibt sich aus dem festzulegenden Parameter ν . Im Vergleich erlaubt diese Aktivierungsfunktion einen schnelleren Trainingsvorgang und entspricht stärker den aus neurowissenschaftlichen Untersuchungen bekannten biologischen Prozessen [9].

Pooling-Schicht: Bildpunkte lassen sich durch die Durchführung einer Unterabtastung innerhalb der jeweiligen rezeptiven Felder ermitteln, was als „Pooling“ bezeichnet wird. Im Rahmen dieser Arbeit werden Pooling-Schichten lediglich mithilfe der Maximumfunktion gebildet. Wenn $R_n(h', w', d', X_n)$ die in einem rezeptiven Feld um den Punkt (h', w', d') enthaltenen Bildpunkte bezeichnet, gilt für die Pooling-Schicht:

$$X_n(h', w', d') = \max(R_{n-1}(h', w', d', X_{n-1}))$$

Eine Unterabtastung kann darüber hinaus durch Mittelwertbildung oder stochastische Abtastung realisiert werden.

Optimierung

Das Trainieren der Netzwerke stellt aus mathematischer Sicht ein Minimierungsproblem dar. Die Ausgänge der letzten Schicht werden dabei als Konfidenzwerte für die betrachteten Kategorien interpretiert, so dass die Dimension der Bildkanäle d der Anzahl der betrachteten Klassen entspricht. Die zunächst beliebig skalierten Konfidenzwerte werden durch Anwendung der Softmax Funktion ρ auf Werte skaliert, welche im Sinne einer Wahrscheinlichkeit interpretiert werden:

$$\rho(h', w', d') = \frac{e^{X_{n-1}(h', w', d')}}{\sum_{k=1}^d e^{X_{n-1}(h', w', k)}}$$

Die multinomiale logistische Verlustfunktion erlaubt es anschließend, das Minimierungsproblem zu definieren.

$$\begin{aligned} \Theta &= \underset{\Theta}{\operatorname{argmin}} E \\ &= \underset{\Theta}{\operatorname{argmin}} \left(-\frac{1}{h \cdot w} \sum_{i=1}^h \sum_{j=1}^w \log(\rho(i, j, l(i, j))) \right) \end{aligned}$$

Dabei bezeichnet $l(i, j)$ die in einem Bildpunkt tatsächlich vorliegende Klasse, so dass die Verlustfunktion für die absolute Konfidenz für die wahre Klasse ihr Minimum annimmt.

Nach Definition der Verlustfunktion erfolgt das Training der CNNs nach dem bekannten Rückwärtspropagierungsverfahren auf Grundlage des Gradientenabstiegs. Praktisch wird die Variante des stochastischen Gradientenabstiegs (SGD) mit Trägheitsterm verwendet. Dieser weist zwei wesentliche Unterschiede zum gewöhnlichen Gradientenabstieg auf. Zum Einen wird der Ausgangsfehler auf einer Teilmenge des Trainingsdatensatzes bestimmt. Der Umfang der Teilmenge wird dabei als „batchsize“ und das vollständige Durchlaufen des Trainingsdatensatzes als „Epoche“ bezeichnet. Zum Anderen wird die Lernregel um einen gewichteten Trägheitsterm erweitert, so dass die Gewichtsanzpassung zusätzlich von der vorausgegangenen Änderung abhängt. Mit der Schrittweite η und der Trägheit α ergibt sich die Lernregel in der Iteration τ damit zu:

$$\Delta\Theta_{n,\tau+1} = -\eta \frac{\partial E}{\partial \Theta_n} + \alpha \Delta\Theta_{n,\tau}$$

Netzwerktopologie

[1] entwickeln eine als „Inception Modul“ bezeichnete Architektur für CNNs, bei welcher Faltungen mit kleinen und großen rezeptiven Feldern ($\hat{=}$ Dimensionen der Faltungsmatrizen) parallel geschaltet werden. Es werden dabei deutlich mehr kleine Faltungsmatrizen verwendet, was durch statistische Überlegungen und biologische Vergleiche auf Grundlage der Hebb’schen Lernregel („neurons that wire together fire together“) begründet wird. Die Ausgänge der parallelen Faltungsschichten werden anschließend entlang der Dimension der Bildkanäle aneinandergelängt. Da im Folgenden die nichtlinearen Aktivierungsschichten

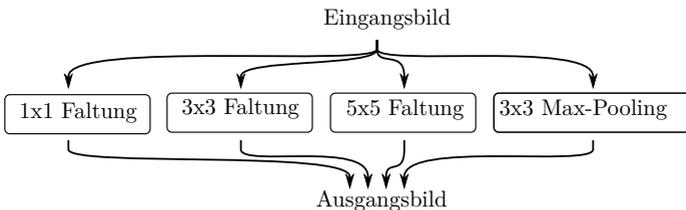


Bild 3: Inception Modul

stets in Kombination mit den Faltungsschichten verwendet werden, wird der Übersichtlichkeit halber auf ihre Darstellung verzichtet, so dass sich die in Abbildung 3 dargestellte Struktur ergibt. Aufbauend auf dieser Struktur wird die als „Google-Net“ bekannt gewordene Netzwerkarchitektur abgeleitet. Diese ist in Abbildung 4 in schwarz dargestellt, wobei der Übersichtlichkeit halber Aktivierungs- und Pooling-Schichten nicht eingezeichnet und hintereinander geschaltete „Inception Module“ zu einem Block zusammengefasst sind. Ursprünglich für die Probleme der Objekt- und Szenenerkennung formuliert, konnten unter Verwendung dieser Modellstruktur gute Benchmarkergebnisse erzielt werden. Bemerkenswerterweise liegt die Anzahl der freien Modellparameter dabei mehr als eine Größenordnung unterhalb der alternativer Netzwerkarchitekturen mit vergleichbarer Genauigkeit. Die „GoogleNet“ Architektur ist nicht ohne weiteres auf das Problem der semantischen Segmentierung anwendbar, da sie im Kontext der ursprünglichen Anwendung eine skalare Ausgangsgröße liefert und hierzu durch Unterabtastungen die räumliche Auflösung der Bildinformationen sukzessive reduziert. Für das Problem der semantischen Segmentierung muss jedoch die räumliche Auflösung prinzipiell unverändert bleiben. Ein Verzicht auf die Unterabtastungen führt auf einen signifikant erhöhten Speicherbedarf und ist deshalb nicht praktikabel. Stattdessen werden, wie in [3] vorgeschlagen, parallele Netzwerkpfade ausgehend von frühen, hochaufgelösten Schichten gebildet. Die entsprechenden Modifikationen an der Netzwerkarchitektur sind in Abbildung 4 mit durchgezogenen grauen Linien veranschaulicht. Die Ausgänge aller parallelen Netzwerkpfade werden zum Gesamtausgang aufsummiert, wobei Pfade mit niedrigerer räumlicher Auflösung zunächst durch eine bilineare Interpolation überabgetastet werden, eine Operation die sich effizient als inverse Faltung oder Dekonvolution implementieren lässt.

Experimentelle Untersuchungen

Alle experimentellen Untersuchungen werden auf Grundlage des CamVid Datensatzes [10] durchgeführt. Dieser enthält 600 Bilder von Verkehrsszenen inklusive ihrer manuell markierten Annotation in 32 verschiedenen Klassen. Nach [11] wird der Datensatz in 367 Trainings- und 233 Testbilder unterteilt und lediglich für die 11 häufigsten Klassen (Straße, Gebäude, Himmel, Baum, Gehweg, PKW, Pfosten, Zaun, Fußgänger, Fahrradfahrer und Verkehrszeichen) ausgewertet. Weiterhin werden die Bilder des Datensatzes während des Trainingsprozesses nicht in der ursprünglichen Auflösung von 960x720 Bildpunkten verarbeitet sondern auf 480x360 Bildpunkte skaliert. Abbildung 1 zeigt beispielhafte Bilder des Datensatzes.

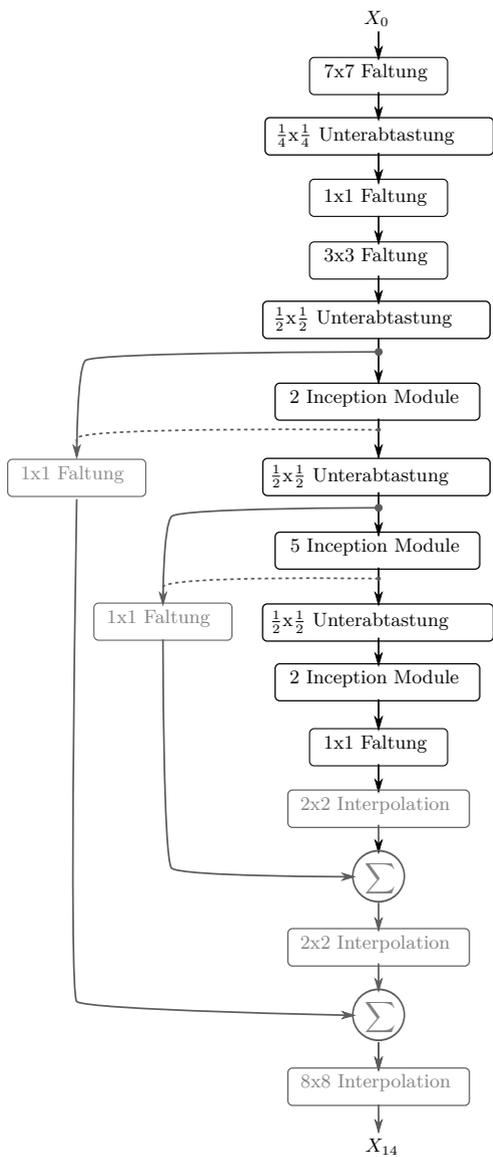


Bild 4: „GoogleNet“ Architektur, Modifikationen in grau dargestellt

Neben der grundsätzlichen Anwendbarkeit des vorgestellten Netzwerkes für das Problem der semantischen Segmentierung, werden im Rahmen der experimentellen Auswertung die getroffenen Architekturentscheidungen näher analysiert. [3] schlägt vor, den Abzweigungspunkt zum Hinzufügen der parallelen, höheraufgelösten Pfade im Netzwerk unmittelbar nach der erfolgten Unterabtastung zu wählen. Tatsächlich existieren im Netzwerk mehrere Abschnitte mit gleichbleibender räumlicher Auflösung der Bildinformation, so dass der Abzweigungspunkt zwischen zwei Unterabtastungen beliebig gewählt werden kann. Im Rahmen der Auswertung wird daher die in [3] vorgeschlagene Architektur, die in Abbildung 4 als durchgezogene, graue Linie gezeichnet ist, mit der Architektur die sich ergibt, wenn die Abzweigungspunkte entsprechend der gestrichelt eingezeichneten grauen Linien gewählt werden verglichen.

Weiterhin schlägt [3] einen schrittweisen Trainingsprozess vor, bei welchem zunächst das ursprüngliche „GoogleNet“ ohne parallele Pfade trainiert wird und die Netzwerkkonstruktion erst nach einer festgelegten Anzahl an Iterationen sukzessive um die parallelen Pfade erweitert wird. Der Einfluss dieses Vorgehens wird analysiert, indem die so erzielten Ergebnisse verglichen werden mit denjenigen Werten, die sich ergeben falls die endgültige Netzwerkkonstruktion von Beginn des Trainingsprozesses an verwendet wird.

Insgesamt ergeben sich hieraus vier verschiedene CNN, welche jeweils für 42000 Iterationen trainiert werden, was mit einer Batchgröße von 8 einer Anzahl von über 900 Epochen entspricht. Die Parametrierung des SGD Algorithmus wird aus [3] übernommen. Für die schrittweise trainierten Modelle werden die parallelen Pfade jeweils nach 14000 und 28000 Iterationen hinzugefügt. Alle Parameter des ursprünglichen „GoogleNet“ (Abbildung 4 schwarz) werden auf die in [1] publizierten Werte initialisiert¹², welche durch das Trainieren auf einem Datensatz zur Szenenerkennung entstanden sind. Nach [3] werden die verbleibenden Faltungsmatrizen des Hauptpfades normalverteilt und diejenigen der parallelen Pfade mit dem Wert Null initialisiert. Die Trainingszeit unter Verwendung des Caffe Frameworks [12] und der cuDNN Bibliothek [13] zur beschleunigten Berechnung der linearen Faltung ergeben sich auf einem Rechnersystem mit GeForce 960 GTX GPU zu ca. 42 Stunden pro Modell, der Bedarf an GPU Speicherplatz bleibt dabei unterhalb von 2GB.

Für die resultierenden Modelle werden mehrere pixelbasierte Metriken bestimmt, darunter die nach [14] als „Accuracy“ bezeichnete Richtig-Positiv-Rate. Weiterhin wird das in [15] angegebene „intersection over union“ (IOU) Maß sowie die F1

¹² Verwendet wird die unter http://caffe.berkeleyvision.org/model_zoo.html öffentlich zugängliche Version (abgerufen am 11. September 2015)

Zahl nach der Definition aus [16] verwendet. Mit den Abkürzungen RP für „richtig positiv“, FP für „falsch positiv“ und FN für „falsch negativ“ gilt:

$$Accuracy = \frac{RP}{RP + FN}$$

$$IOU = \frac{RP}{RP + FP + FN}$$

$$F1 = 2 \frac{\frac{RP}{RP+FP} * \frac{RP}{RP+FN}}{\frac{RP}{RP+FP} + \frac{RP}{RP+FN}}$$

Tabelle 1: Ergebnisse der vier CNNs: der Verzweigungspunkt ist mit „early/late“, das Training mit „at once/step-wise“ gekennzeichnet

	Straße	Gebäude	Himmel	Baum	Gehweg	PKW	Pfosten	Zaun	Fußgänger	Fahrradfahrer	Verkehrszeichen	Global pro Pixel	Global pro Klasse
Acc.	%	%	%	%	%	%	%	%	%	%	%	%	%
eo	92,54	92,48	94,96	81,79	45,24	68,95	28,84	32,19	57,75	10,15	2,09	83,73	57,56
es	96,43	92,37	94,37	82,20	45,04	67,64	28,06	35,07	60,16	18,18	10,94	84,69	59,59
lo	97,48	92,44	95,03	78,77	41,87	73,57	31,14	41,77	55,25	26,60	12,32	84,71	60,91
ls	96,65	93,05	94,59	79,33	43,90	73,42	27,90	39,88	64,15	28,34	15,28	84,83	61,78
IOU	%	%	%	%	%	%	%	%	%	%	%	%	%
eo	76,04	80,81	89,44	65,04	41,19	65,22	16,19	28,65	41,29	9,87	2,07	83,73	49,96
es	78,73	80,73	89,51	68,44	41,70	63,79	16,89	32,25	43,36	16,77	10,17	84,69	52,25
lo	78,66	80,62	89,62	67,83	39,28	69,62	18,42	36,55	42,10	23,84	12,01	84,70	53,60
ls	78,70	80,89	89,57	67,56	40,99	67,34	18,53	34,47	46,32	25,14	13,51	84,83	53,99
F1	%	%	%	%	%	%	%	%	%	%	%	%	%
eo	86,39	89,39	94,43	78,82	58,35	78,95	27,86	44,55	58,45	17,96	4,06	91,14	60,86
es	88,10	89,34	94,46	81,26	58,85	77,89	28,90	48,77	60,50	28,72	18,46	91,71	63,91
lo	88,06	89,27	94,52	80,83	56,40	82,09	31,11	53,54	59,25	38,50	21,44	91,72	65,56
ls	88,08	89,44	94,50	80,64	58,15	80,48	31,26	51,26	63,32	40,18	23,81	91,79	66,08

Im einzelnen ergeben sich die in Tabelle 1 aufgeführten Gütemaße, falls der Testdatensatz in voller Auflösung ausgewertet wird. Die Ergebnisse deuten darauf hin, dass die in [3] vorgeschlagene Wahl der Verzweigungspunkte auf dem CamVid Datensatz nicht die optimalen Ergebnisse liefert. Wird das Netzwerk stattdessen in späteren Schichten verzweigt (gestrichelte Linien in Abbildung 4), lassen sich die Ergebnisse je nach betrachtetem Gütemaß um teilweise mehrere Prozentpunk-

te steigern. Die Modelle, welche schrittweise trainiert werden, erzielen in den globalen Gütemaßen bessere Ergebnisse, wobei das schrittweise Training weniger Einfluss als die Wahl des Verzweigungspunktes hat. Bei der Betrachtung der auf die Kategorien aufgeschlüsselten Ergebnisse fällt auf, dass insbesondere Kategorien von Verkehrselementen, die jeweils nur kleine Flächen im Bild einnehmen, mit einer vergleichsweise niedrigen Genauigkeit erkannt werden. Dies ist kaum verwunderlich, da die Ausgangsbilder trotz hinzufügen der parallelen Netzwerkpfade durch Interpolation von achtfach unter-abgetasteten Bildinformationen bestimmt werden.

Abbildung 5 zeigt die Segmentierungen aller Modelle für drei beispielhafte Bilder des Testdatensatzes. In diesen Bildern fällt auf, dass es in nur wenigen Fällen gelingt, Gehwege korrekt zu segmentieren. Eine Ursache für diese Fehlklassifikation könnte darin liegen, dass sich Gehwege und Straßen in ihrer lokalen Erscheinung kaum voneinander unterscheiden. Dies verdeutlicht einen bedeutenden Nachteil der lokal stark begrenzten rezeptiven Felder, welche globale Anhaltspunkte wie etwa die Lage der Verkehrselemente im Bild, nicht ausreichend berücksichtigen.

Zusammenfassung und Ausblick

Die vorliegende Arbeit untersucht, wie sich die semantische Kategorie als diskrete Eigenschaft jedes Punkts eines Kamerabilds mittels Convolutional Neural Networks klassifizieren lässt. Hierzu wird ein System auf Grundlage der GoogleNet Architektur beschrieben, welches um parallele Netzwerkpfade ausgehend von frühen, höher aufgelösten Schichten mit geringem zusätzlichem Rechenaufwand erweitert wird. Die experimentelle Verifikation erfolgt anhand realer Aufnahmen des etablierten CamVid Datensatzes. Die Ergebnisse bestätigen die grundsätzliche Eignung des vorgestellten Ansatzes zur Bestimmung einer semantischen Segmentierung. In zukünftigen Arbeiten ist die Genauigkeit der semantischen Segmentierung durch eine zeitliche Fusion der aufgezeichneten Kamerabilder weiter zu verbessern. Weiterhin erfordern Anwendungen in realen Fahrzeugen eine geometrische Rekonstruktion der Verkehrsszene, welche aufbauend auf einer semantischen Segmentierung gebildet werden kann.

Danksagung

Die Autoren danken Carlos Miguel Treviño Campa für seine Unterstützung bei der Durchführung der experimentellen Untersuchungen.

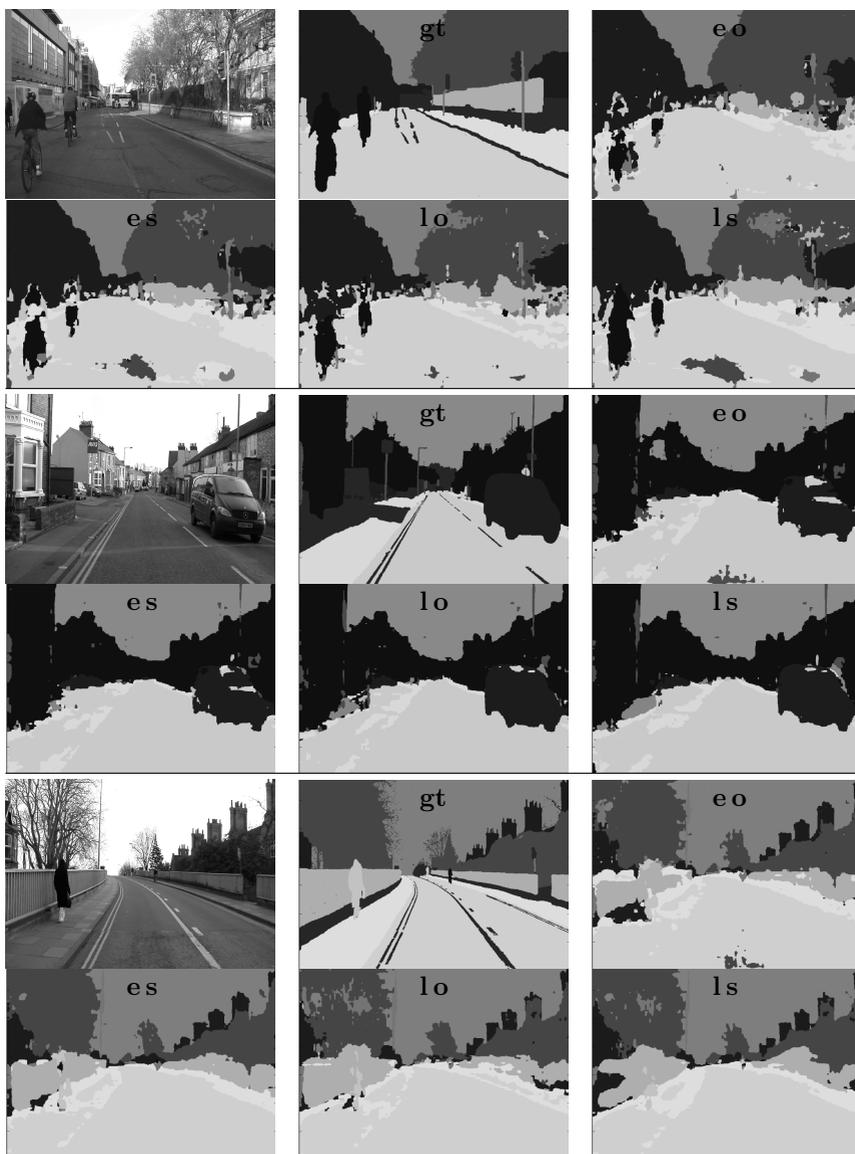


Bild 5: Beispielergebnisse der vier CNNs, „gt“ markiert die realen Annotationen

Literatur

- [1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich. „Going Deeper With Convolutions“. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (S. 1-9). 2015.
- [2] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell. „DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition“. In: *Proceedings of The 31st International Conference on Machine Learning* (S. 647-655). 2014.
- [3] J. Long, E. Shelhamer, T. Darrell. „Fully Convolutional Networks for Semantic Segmentation“. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (S. 3431-3440). 2015.
- [4] G. Lin, C. Shen, I. Reid, I. (2015). „Efficient piecewise training of deep structured models for semantic segmentation“. *arXiv preprint arXiv:1504.01013*. 2015
- [5] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille. „Semantic image segmentation with deep convolutional nets and fully connected crfs“. *arXiv preprint arXiv:1412.7062*. 2014
- [6] C. Farabet, C. Couprie, L. Najman, Y. LeCun. „Learning hierarchical features for scene labeling“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8) (S. 1915-1929). 2013
- [7] S. Hong, H. Noh, B. Han. „Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation“. *arXiv preprint arXiv:1506.04924*. 2015
- [8] B. Hariharan, P. Arbeláez, R. Girshick, J. Malik. „Hypercolumns for object segmentation and fine-grained localization“. *arXiv preprint arXiv:1411.5752*. 2014
- [9] X. Glorot, A. Bordes, Y. Bengio. „Deep sparse rectifier neural networks“. In: *International Conference on Artificial Intelligence and Statistics* (S. 315-323). 2011
- [10] G. J. Brostow, J. Fauqueur, R. Cipolla. „Semantic object classes in video: A high-definition ground truth database“. *Pattern Recognition Letters* 30(2) (S. 88-97). 2009

- [11] G. J. Brostow, J., Shotton, J. Fauqueur, R. Cipolla. „Segmentation and recognition using structure from motion point clouds“. In: *Proceedings of the 10th European Conference on Computer Vision: Part I* (S. 44-57). 2008
- [12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell „Caffe: Convolutional Architecture for Fast Feature Embedding“. In: *Proceedings of the ACM International Conference on Multimedia* (S. 675-678) 2014
- [13] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, E. Shelhamer. „cudnn: Efficient primitives for deep learning“. *arXiv preprint arXiv:1410.0759*. 2014
- [14] B. Liu, X. He, S. Gould. „Multi-class semantic video segmentation with exemplar-based object reasoning“. In: *IEEE Winter Conference on Applications of Computer Vision* (S. 1014-1021). 2015
- [15] M. Cogswell, X. Lin, S. Purushwalkam, D. Batra. „Combining the best of graphical models and convnets for semantic segmentation“. *arXiv preprint arXiv:1412.4313*. 2014
- [16] J. Fritsch, T., Kuhl, A. Geiger. „A new performance measure and evaluation benchmark for road detection algorithms“. In: *16th International IEEE Conference on Intelligent Transportation Systems* (S. 1693-1700). 2013

Analyzing Capabilities of Latin Hypercube Designs Compared to Classical Experimental Design Methods

Andreas Fischbach, Jörg Stork, Martin Zaefferer, Sebastian Krey,
Thomas Bartz-Beielstein

SPOTSeven Lab, Dept. of Comp. Sci. and Eng. Sci.
TH Köln
E-Mail: firstname.lastname@th-koeln.de
www.spotseven.de

Introduction

Design of experiments (DOE) has proven to be a useful tool to optimize process or production output [1]. A design specifies which values for input parameters of an experiment are to be chosen to reach a desired output or gather a maximum amount of information.

Many authors recommend space-filling and non-collapsing designs for deterministic computer experiments [2, 3]. *Latin hypercube designs* (LHDs) are non-collapsing by default due to their creation rules. The space-filling property can be fulfilled by maximizing the minimum Euclidean distance of points. Those designs are especially useful for fitting Kriging models [4]. But it has not been proven, if they are the best choice [2]:

In sum, it has not been demonstrated that LHDs are superior to any designs other than simple random sampling (and they are only superior to simple random sampling in some cases).

Based on the experimental analysis of Santner [2], this work analyses the behavior of different LHD types (space-filling and non-space-filling) used to fit Kriging metamodels and compare their properties and performance to classical design of experiment methods.

Designs can be evaluated with regards to optimality. For a given model, information based statistical criteria like D-optimality (maximize the determinant of the information matrix), A-optimality (minimize the trace of the inverse of

the information matrix) and E-optimality (maximize minimum eigenvalue of the information matrix) can be calculated.

Searching for an optimal design even with the search space limited to a subclass of all designs is still difficult [5]. Often, a search for an optimal design is stopped before convergence, to avoid spending too much time on design generation. So, in this work the approach is to randomly draw designs instead of creating optimized designs with much effort. Afterwards the drawn designs are evaluated and compared with optimized designs to determine, which properties and criteria a design must fulfill to be adequate for fitting Kriging metamodels.

This work compares model dependent properties like the mentioned optimality criteria with spatial criteria like minimum and average Euclidean interpoint distance. The goal of this paper is to find answers to the following research questions:

- Which criteria must a design fulfill to enable fitting high quality Kriging metamodels?
- Does the design type have a significant impact on the model quality?
- How many points should a good design consist of (regarding the tradeoff between cost intensive number of experiments and the model quality)?

In this work, the designs' properties and their effect on building Kriging models will be observed using a simple test function. This enables a very transparent and easy to reproduce comparison. Also, sufficiently accurate models may be build with a very limited number of design points.

Methods

Optimality Criteria

The optimality criteria observed in this work can be divided in two groups. The first contains model dependent criteria and the second spatial criteria with no reference to an underlying (surrogate) model. The model-dependent criteria are computed based on the information matrix of each design corresponding to a simple first order linear regression model [6].

Model-dependent Criteria

Consider the linear model in matrix form, $\mathbf{Y} = \beta\mathbf{X} + \epsilon$, where the vector ϵ contains the n random errors (n denotes the sample size). The elements of the error vector, ϵ_i , are assumed to be independent and identically normally distributed with mean zero and error variance σ_ϵ^2 . The vector of unknown coefficients, β , can be estimated via least squares methods as $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{Y}$. Its covariance matrix is $\text{var}(\hat{\beta}) = \sigma_\epsilon^2(\mathbf{X}^T\mathbf{X})^{-1}$. The square roots of the diagonal entries of this matrix are the standard errors of the β_i 's. The diagonal elements of $(\mathbf{X}^T\mathbf{X})^{-1}$ are called the relative variances of the β_i 's, $v_i = \text{var}(\hat{\beta})/\sigma_\epsilon^2$. The v_i 's indicate how large the variances of the estimated model parameters compared to the error variance σ_ϵ are. The inverse of the covariance matrix $\text{var}(\hat{\beta})$ is called the *information matrix* for the model parameter vector $\hat{\beta}$, i.e., $\mathbf{M} = 1/\sigma_\epsilon^2\mathbf{X}^T\mathbf{X}$. The information matrix is a measure of the information about the factor effects that is contained in the design. The information matrix measures the information on factor effects that is contained in the design.

An experimental design that maximizes the determinant of the information matrix, $|\mathbf{M}|$, is called *D-optimal*. The minimization of $|(\mathbf{X}^T\mathbf{X})^{-1}|$ is equivalent to the maximization of $|(\mathbf{X}^T\mathbf{X})|$.

A-optimality minimizes the trace of the inverse of the information matrix and is often used as a criterion for designs with equal D-optimality.

E-optimality maximizes the smallest eigenvalue of the information matrix. The aim of E-optimality is to minimize the maximum variance of all possible normalized linear combinations of parameter estimates. E-optimal designs minimize the maximum axis of the confidence ellipsoid of estimators, namely, E-optimal designs minimize the maximum eigenvalue of the covariance matrix of estimators.

An overview of these criteria is given in [7]. The relationship between these three optimality criteria is illustrated in [8].

Spatial Criteria

The spatial criteria analyzed in this work are the Cartesian pairwise point distances of a design. Let m denote the problem dimension and $d_{ij} = d(X_i, X_j) = (\sum_{k=1}^m |X_{ik} - X_{jk}|^p)^{1/p}$, with $p \in \{1, 2\}$, be the *distance* between two *design points* X_i and X_j . A *maximin* distance design maximizes the minimum pairwise point distance, i.e., $d_{\min} = \min_{1 \leq i, j \leq n, i \neq j} d(X_i, X_j)$. Additional optimization criteria for LHDs are discussed in [5]. *Minimax* distance designs are defined in a similar manner [9].

Quality Criteria

In the following we will analyze the model quality for each design fitting a Kriging model and measure the impact of the mentioned design properties, namely

- minimum distance: d_{\min} ,
- average distance: d_{avg} ,
- optimality: A-, D-, and E.

Note, not every criterion from this list is used as an optimality criterion during the design generation. For example, the average distance will be used only to visualize and analyze results. Our analysis is based on results obtained by Santner [2].

Design types

To answer the previously stated questions several experimental designs will be generated. The below described designs are divided in different types, namely *Maximin LHD*, *Degenerated LHD*, *Optimized LHD* and *Uniform*. All types have in common that their designs are created based on pseudo random number sampling. Therefore experiments using these designs are repeated multiple times and statistically analyzed to lead to robust results.

The *Maximin LHD* uses pairwise Cartesian distances of the design points to maximize the minimum distance and thus lead to a space filling design. Out of 1,000 randomly created LHDs the one with the maximum minimum distance will be taken.

Degenerated LHD tries to compute a non space-filling Latin hypercube design to be able to analyze the importance and necessity of the space-filling property. For this type the design out of 1,000 randomly generated LHDs with the minimum sum of the pairwise Euclidean distances will be drawn. An *Optimized LHD* is computed using the simulated annealing algorithm to optimize an LHD via the Φp criterion, which is linked to the minimum distance criteria [10]. Regarding computation time, especially for large designs, this design type is computationally very cost intensive.

The various LHDs are compared to a simple random sampling approach. Here, design points are sampled from a uniform distribution, hence this approach is denoted *Uniform*. The comparison of *LHD* and *Uniform* approaches provides a simple benchmark. Furthermore, it allows to determine the necessity of the space-filling and non-collapsing property of the LHDs

To get an idea if and how classical designs are also applicable for computer experiments and if they are able to outperform LHDs when fitting Kriging-models, *Full Factorial Designs* (FFD) with appropriate number of design points and additional center points are also included in the experiments.

Metamodel

Kriging (or Gaussian Process Regression) is a frequently used surrogate-model. It is an excellent predictor of smooth, continuous problem landscapes. Moreover, it provides an uncertainty estimate of its own prediction, which can be used to calculate the *Expected Improvement* (EI) of a candidate solution. EI is one great benefit of Kriging models and is for example used in the Efficient Global Optimization algorithm introduced by Jones et al. [11] to balance exploitation and exploration in a Kriging-based optimization process.

The Kriging implementation used in the experiments is taken from the SPOT R-Package¹³. This implementation is based on earlier code by Forrester et al. [12], who also provide a very comprehensive description of Kriging and Kriging-based optimization.

A Kriging model with default parameters is used. The quality of a design is computed by evaluating the *Root Mean Squared Error* (RMSE) of a derived Kriging model. For that purpose, the residuals of the model on an equidistant grid are computed.

Experiments

The described experiments are based on work done by Santner in 2003 [2]. Santner compared the behavior of a space-filling and evenly spread LHD with a degenerated LHD with design points at the diagonal axis of a unit square.

Objective Function

The experiments were conducted using the objective function defined in Santner [2]:

$$y(x_1, x_2) = \frac{x_1}{1 + x_2} \quad (x_1, x_2) \in [0, 1] \times [0, 1]. \quad (1)$$

¹³ See <https://cran.r-project.org/package=SPOT>

This objective function is relatively easy to model (see Figure 1). It allows to focus on the behavior of the design according to its properties instead of the modeling method itself. The surface of the function is almost plain showing a linear trend and there is an interaction between x_1 and x_2 .

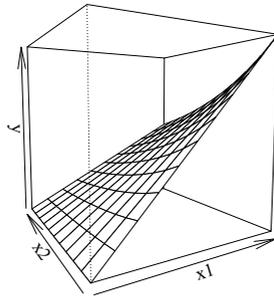


Figure 1: Objective Function used in all Experiments.

Setup

30 random instances of the previously described design types are drawn for design sizes from $n = 5$ to $n = 12$ design points. For $n = 5$ and $n = 9$ a FFD design is added resulting in a total number of 962 simulation experiments ($30 \times$ number of design types \times number of different design sizes $+ 2$ FFD experiments). In each experiment the properties of the used design are computed (pairwise Euclidean distances and A-, D- and E-optimality).

For each design, a Kriging model of the objective function is fitted using the function values, y_i , at the n design points X_i . Afterwards the model fit is used to predict values at a pre-defined grid over the region of interest from 0.05 to 0.95 in each dimension with a distance of 0.1. These values are referred to as \hat{Y}_i . This results in $l = 100$ prediction values that are used to compute the difference to the objective function values at the grid points. With these values the RMSE as defined in Equation 2 is computed.

$$\text{RMSE} = \sqrt{\frac{1}{l} \sum_{i=1}^l (y_i - \hat{Y}_i)^2}. \quad (2)$$

To answer the question if there is a correlation between the model based optimality criteria of the designs and the quality of the fitted Kriging models, the optimality criteria are computed using the design information matrix \mathbf{M} with regard to a linear regression model. The model contains terms for each main effect x_1 and x_2 and the interaction term $x_1 * x_2$.

Results

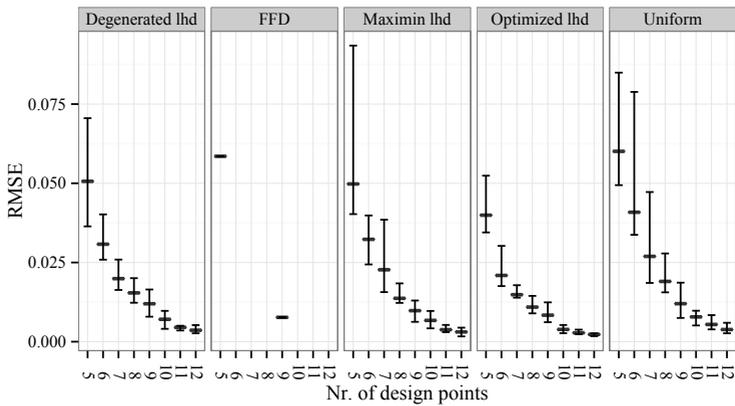


Figure 2: Development of the model error per design type by increasing number of design points.

An overview of the results from the experiments is given in Figure 2. The boxplot shows the median value and the upper and lower quartiles of the RMSE of a set of experiments. The values are grouped by the design type and the number of design points. The column FFD shows only two results due to the fact that the Full Factorial Designs with center points are deterministic. It can be seen, that model error and variance of error decrease with increasing number of design points. Although there are further improvements to be expected when design points are added, a convergence for each type is tendentially recognizable. Further increasing of the design size would not lead to a significantly improved model performance.

In the FFD column the experiment results of the two single experiments for five and nine points are shown. The five point FFD performance is similar to

the median of the other five point designs. On the other hand, the nine point FFD's performance seems to be better than the median performance of the other types.

The performance differences of the design types can best be seen with a minimum number of design points, say five to seven points. The median RMSE of the Uniform designs are clearly worse and the variance is largest. It is somehow surprising that the *Degenerated LHD* seems to perform at the same level compared to the others, except of *Uniform*. In sum it can be stated, that it is not so easy to consistently create bad performing non-space-filling LHDs.

Table 1: Analysis of Variance of experiment results

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
type	3	0.0232	0.0077	19.1479	4.63e-12	***
nrDesignPoints	1	0.3035	0.3035	751.1904	< 2e-16	***
minDist	1	0.0004	0.0004	0.8743	0.350014	
avgDist	1	0.0056	0.0056	13.8912	0.000205	***
dOptimality	1	0.0115	0.0115	28.5166	1.16e-07	***
aOptimality	1	0.0001	0.0001	0.3022	0.582629	
eOptimality	1	0.0087	0.0087	21.6129	3.81e-06	***
Residuals	950	0.3838	0.0004			

To conclude if the design type among some or all of the designs properties have a significant impact on the model error an ANOVA is performed. The two experimental results of the *Full Factorial Designs* for five and nine design points are removed in advance.

Table 1 shows the results of the ANOVA. The analysis shows that the type has a significant effect on the model error. Besides the type it can be seen, that the number of design points (as expected), the average Euclidean distance, D-optimality and E-optimality have a significant impact at a 99% significance level. Regarding the mean of squares the number of design points has obviously the largest impact on the model error. The other significant parameters seem to influence the model error at almost the same level.

To further analyze and compare the different used design types and to answer the question which design type works best a Tukey's honest significance test is performed to present a pairwise comparison of the different design types. The results are shown in Figure 3. The *Degenerated LHD* and *Maximin LHD* perform on an equal level. Also the *Optimized LHD* and the *Degenerated LHD* perform

not significantly different, the same holds for the *Optimized LHD* and *Maximin LHD*. All LHD types are superior to *Uniform* in performance at a 95% confidence level.

One stated question is which properties of the described experimental designs have a significant impact on the model quality. The ANOVA results (see Table 1) shows the significance of the design properties. Figures 4 to 6 show the development of the resulting RMSE for increasing values of the minimum distance, D-optimality and A-optimality grouped by the number of design points. The solid black line shows a trend of the RMSE for the parameter, computed via local smoothing of the collected data. In these results, the two experiments with the *FFD* are included.

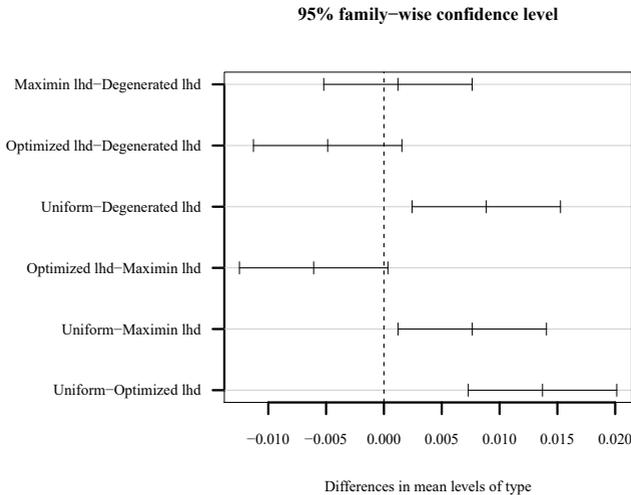


Figure 3: Pairwise differences of model error for LHD types at 95% confidence level. This figure illustrates, e.g., that Uniform Designs (random sampling) is worse compared to Optimized LHD, because the corresponding confidence interval does not contain zero. It is positive, so the difference in the RMSE values is positive, too.

Regarding the ANOVA results, a look at the trend of the *minimum distance* parameter (see Figure 4) also shows a decreasing RMSE by increasing the minimum distance, but not consistently for all number of design points and additionally the effect is decreasing with increasing design size.

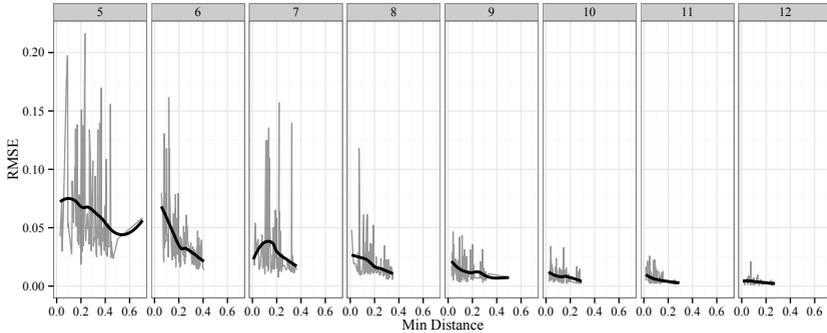


Figure 4: Effect of minimum distance property d_{\min} on model error grouped by number of design points.

D-optimality is often analyzed together with *A-optimality*. If two or more designs with an equal value for D-optimality are found, the experimenter often chooses the design with the best value for *A-optimality*. Figure 5 clearly shows the positive effect of *D-optimality* on the model RMSE. In contrast to the general trend, rather high RMSE values can be observed for the rarely occurring large D-optimality values.

It also can be seen that with larger designs the value range for the determinant (printed at a logarithmic scale) is decreasing and lower values occur less frequent. Figure 6 shows the development of the *A-optimality*, which is desired to be maximized. A positive impact of A-optimality on the RMSE can be recognized, especially for seven and nine design points. This is observed despite of the fact that the earlier described ANOVA evaluated the impact of A-optimality to be not significant.

Regarding the researched question how many points a design should consist of at least, it can be seen that for five design points the variance of the results is quite large. Slightly increasing the design size up to seven points leads to better models and reduced variance in the RMSE (see Figure 2). The *Optimized LHD* seems to be the most robust design type.

The best five designs out of 120 for a design size of $n = 5$ and $n = 9$ points with their ranked properties are given in Table 2 and 3. For a design size of five it can be seen that the *Optimized LHD* leads to good ranked values (third and sixth best values) for minimum distance but not automatically to the best RMSE. The

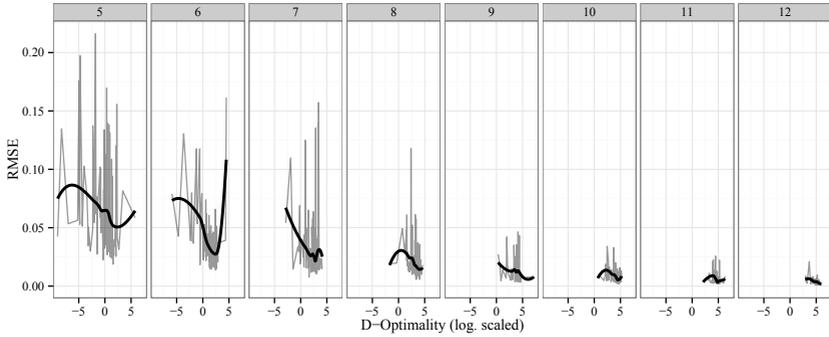


Figure 5: Effect of D-optimality, $|\mathbf{M}|$, on model error grouped by number of design points.

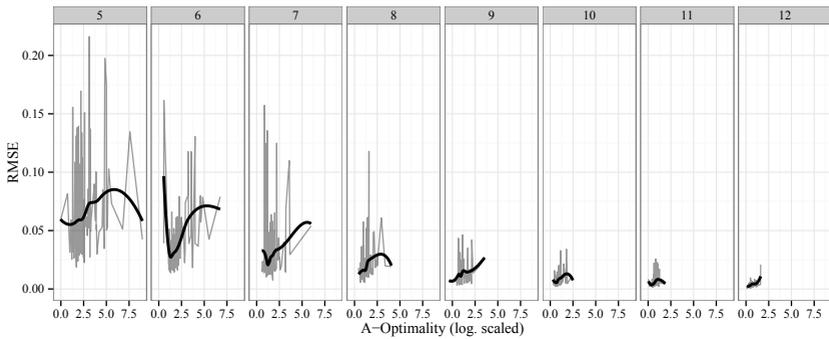


Figure 6: Effect of A-optimality, $\text{trace}(\mathbf{M})$, on model error grouped by number of design points.

best design, a *Degenerated LHD* has no property resulting in a top ten rank, but good values for D-optimality (17th), A-optimality (26th) and E-optimality (35th). For nine design points at least the best design is an *Optimized LHD*, but the RMSE values of the following designs differ not much. The best designs properties yield good ranks (7th and 8th) for D-, A- and E-Optimality. The third best design is to be mentioned because the computed properties are all ranked around the 90th place out of 120. So it seems worth to not only focus on one single criterion to optimize when looking for suitable designs for fitting Kriging metamodels.

Table 2: Best performing designs and ranked properties for n=5

type	RMSE	minDist	avgDist	D-opt	A-opt	E-opt
Degen. lhd	0.0187	87	41	17	26	35
Maximin lhd	0.0225	86	104	84	78	78
Optim. lhd	0.0237	3	56	24	38	44
Optim. lhd	0.0244	6	30	27	20	21
Degen. lhd	0.0253	46	82	31	29	31

Table 3: Best performing designs and ranked properties for n=9

type	RMSE	minDist	avgDist	D-opt	A-opt	E-opt
Optim. lhd	0.0031	16	38	7	7	8
Maximin lhd	0.0035	94	15	35	36	38
Maximin lhd	0.0036	98	97	92	87	84
Degen. lhd	0.0038	62	8	14	8	7
Degen. lhd	0.0041	55	30	43	59	61

Conclusion

This work analyzes the behavior of different design of experiment methods and the design properties including optimality criteria. The focus is on determining whether these properties and criteria have an influence on the quality of a Kriging model. The research covers the questions of the impact of the design type, the significance of model dependent optimality criteria and spatial criteria and the performance development of the designs by increasing design size.

The experiment results show that LHDs are superior to designs created by simple random sampling, especially at a small number of design points. *Optimized LHDs* are more robust and perform equal to *Maximin LHDs* and *Degenerated LHDs*. Hence the usage of them can be advantageous if the situation allows to afford the computation time. If computation time is an important aspect *Maximin LHDs* should be sufficient.

It seems that the average distance of the design points can be more important than the minimum distance. Further studies are required to verify that thesis. A positive effect of model dependent optimality criteria (for first order linear regression models) can be recognized, although the effect is not significant for all computed criteria. Regarding the results of the FFD experiments, it seems interesting to combine the abilities of standard designs with LHDs.

- **Which criteria must a design fulfill to enable fitting high quality Kriging metamodels?**

A general answer to this question was not found, but maximizing the minimum interpoint distance does not automatically lead to the best designs. Similar to E-optimality, the smallest eigenvalue of a correlation matrix of a Kriging model may be maximized. This may provide an interesting approach towards generating optimal designs for Kriging models.

- **Does the design type have a significant impact on the model quality?**

Simple random sampling lead to significant worse designs. But it is also not easy to generate consistently bad performing LHDs, especially for larger designs. Surprisingly the different examined LHD types perform all on an equal level. This might be due to the simplistic experiment setup. A more sophisticated experimental analyses might yield deeper insights to effects of different design types.

- **How many points should a good design consist of (regarding the tradeoff between cost intensive number of experiments and the model quality)?**

The answer to this question depends on the structure and complexity of the objective function and the expected costs of an experiment. In this case designs consisting of at least seven points lead to a significant decrease in the variance and the median values of the model error. For design sizes larger than seven points the improvement of the model quality becomes less significant.

Interesting further studies include the reproduction of the results on different more complex test functions created by a randomized function generator. The work should be extended to find statistical robust criteria for optimal designs for Kriging models.

Acknowledgements

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Wirtschaft und Energie unter den Förderkennzeichen KF3145101WM3 und KF3145103WM4 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

References

- [1] W. Kleppmann, *Versuchsplanung*. Hanser, 2013.
- [2] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*. Berlin, Heidelberg, New York: Springer, 2003.
- [3] B. Husslage, G. Rennen, E. van Dam, and D. den Hertog, “Space-filling latin hypercube designs for computer experiments,” *Optimization and Engineering*, vol. 12, no. 4, pp. 611–630, 2011.
- [4] J. P. C. Kleijnen, *Design and analysis of simulation experiments*. New York NY: Springer, 2008.
- [5] R. Jin, X. Du, and W. Chen, “The use of metamodeling techniques for optimization under uncertainty.” *Journal of Structural & Multidisciplinary Optimization* (in press), 2005.
- [6] D. C. Montgomery, *Statistical Quality Control*. Wiley, 2008.
- [7] H. Bandemer and A. Bellmann, *Statistische Versuchsplanung*. Teubner, 1994.
- [8] T. Takeuchi and H. Sekido, “An Approximate Approach to E-optimal Designs for Weighted Polynomial Regression by Using Tchebycheff Systems and Orthogonal Polynomials,” *ArXiv e-prints*, Mar. 2013.
- [9] S. Crary, “Design of computer experiments for metamodel generation,” *Analog Integrated Circuits and Signal Processing*, vol. 32, no. 1, pp. 7–16, 2002.
- [10] M. D. Morris and T. J. Mitchell, “Exploratory designs for computational experiments,” *Journal of Statistical Planning and Inference*, vol. 43, pp. 381–402, Feb. 1995.

- [11] D. Jones, M. Schonlau, and W. Welch, “Efficient global optimization of expensive black-box functions,” *Journal of Global Optimization*, vol. 13, pp. 455–492, 1998.
- [12] A. Forrester, A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling*. Wiley, 2008.

Function Generator Application: Shall Corners be Measured?

J. Belz, O. Nelles

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

Introduction

We propose a polynomial-based function generator to support decision-making in the context of experimental modeling (identification). The function generator tries to imitate regression problems in engineering applications. Stochastic elements ensure high variability between generated functions, while the user is able to choose a general complexity level defined by the strength of the nonlinearity and the order of interactions. An extension to overcome unfavorable properties of the polynomial-based structure is made. The ability to generate an arbitrary amount of test functions offers the possibility to statistically secure decisions in the development of algorithms or for the modeling task at hand. To demonstrate the abilities of our proposed function generator, it is utilized to pick a strategy for the design of experiments that should be used for the metamodeling of a centrifugal fan. We show, that for the application at hand the inclusion of all corners in the experimental design is destructive for the meta model's generalization performance.

Requirements

During the development of new algorithms in machine learning and for demonstration of their strengths and weaknesses, an arbitrary amount of synthetic examples would be of great help.

In contrast to typical real-world or simulation data of specific applications, this offers the possibility to easily vary important factors such as:

- amount of data,
- dimensionality (number of inputs),
- strength of nonlinearity,
- data distribution,
- noise level, etc.

Furthermore the generation of multiple data sets of arbitrary size, e.g., for training, validation, testing, with arbitrary input data distributions is no problem. Design of experiments and active learning strategies can be investigated nicely which is completely impossible for fixed data sets.

Of course any structure present in a function generator will favor certain algorithms over others. This paper proposes a polynomial-based approach. Thus any polynomial model-based algorithm will have an unfair advantage. Therefore the polynomial model can be transformed in a second step by a nonlinear function such as a sigmoid.

Infinite possibilities exist for building a function generator and there exists no overall “best” solution. The authors of this paper have mainly engineering applications in mind. A crucial point for any function generator is to offer a wide range of nonlinear characteristics controlled by very few parameters.

Unfortunately extremely few proposals for a function generator can be found in the literature. One very primitive approach has been made in the context of the massive online analysis (MOA) project [1] which is based on a random radial basis function network. A much more sophisticated approach proposed by Friedman [5] is based on additive Gaussians which gives an unfair advantage to all algorithms utilizing Gaussian kernels (typically used in support vector machines, radial basis function networks, Gaussian process models, etc.). In general, care must be taken to recognize a function generator’s underlying structure. A vision for the future is to combine many types of function generators to be more robust in this respect.

Approach

The input is normalized and thus shall vary in the unit hypercube $[0, 1]^n$.

Polynomials

As an n -dimensional nonlinear function the following M -term polynomial is proposed:

$$g(u_1, u_2, \dots, u_n) = \sum_{i=1}^M c_i \cdot (u_1 - s_{i1})^{p_{i1}} \cdot \dots \cdot (u_2 - s_{i2})^{p_{i2}} \cdot \dots \cdot (u_n - s_{in})^{p_{in}}. \quad (1)$$

Its complexity can be controlled via two parameters: the number of terms M and the average magnitude μ of the exponents $p_{i1}, p_{i2}, \dots, p_{in}$ (see below). It is proposed to draw the coefficients c_i from a normal distribution $N(0, 1)$ and the shifts s_{ij} from a uniform distribution $U[0, 1]$. The latter ensures that all the bases $u_j - s_{ij}$ are in the range $[-1, 1]$. Thus this range is kept after raising to the powers independent of the magnitude of the exponents p_{ij} .

The powers p_{ij} are non-negative values. To choose the exponents of the indeterminates (shifted inputs) an exponential distribution with expected value μ is recommended:

$$f(x|\mu) = \frac{1}{\mu} e^{-x/\mu}. \quad (2)$$

The exponents are generated by taking the floor of values drawn from this distribution. The expected value μ is a design parameter that determines

- the probability for high exponents and
- the probability for higher-order interactions because exponents equal to zero make an input irrelevant for a certain term.

Thus the strength of the nonlinearity and the order of interactions are coupled. Both characterize a function's complexity. The frequent zeros resulting from taking the floor of an exponential distribution limit the order of interactions to a reasonable amount. Nevertheless the maximum order of interactions grows with the dimensionality of the problem. For high dimensions it might be reasonable to introduce an additional third parameter in order to control both effects separately. Literature consistently claims that for most real-world problems very-high-order interactions tend to be scarce and not dominant, e.g. [7, 12] (Sect. 10.11). This is

sometimes referred to as the *sparsity-of-effects principle* [3]. A typical realization for the exponents $p_{1j}, p_{2j}, \dots, p_{Mj}$ for $M = 10$ terms (rows) in $n = 8$ dimensions (columns) with expected value $\mu = 1$ (containing 2 – 4-order interactions) is

$$\begin{pmatrix} 0 & 0 & 2 & 0 & 2 & 0 & 0 & 3 \\ 5 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 2 & 0 \\ 1 & 0 & 1 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 3 & 2 \\ 0 & 0 & 1 & 0 & 0 & 2 & 0 & 2 \\ 1 & 0 & 2 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 1 & 6 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \tag{3}$$

and with $\mu = 2$ (containing 3 – 6-order interactions)

$$\begin{pmatrix} 0 & 5 & 1 & 0 & 1 & 0 & 6 & 1 \\ 1 & 2 & 0 & 1 & 2 & 1 & 0 & 2 \\ 0 & 2 & 5 & 0 & 0 & 4 & 1 & 0 \\ 0 & 2 & 0 & 0 & 1 & 0 & 0 & 9 \\ 4 & 2 & 0 & 1 & 1 & 2 & 0 & 0 \\ 2 & 1 & 0 & 3 & 4 & 2 & 2 & 3 \\ 1 & 0 & 1 & 4 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 6 & 1 & 0 \\ 0 & 2 & 1 & 0 & 1 & 0 & 1 & 5 \\ 3 & 0 & 6 & 1 & 4 & 3 & 1 & 0 \end{pmatrix}. \tag{4}$$

The Figs. 1–3 show six two-dimensional polynomial example functions with ten terms each for $\mu = 1, 2, 3$.

Saturated Polynomials

Polynomial functions exhibit at least two unfavorable properties:

- The slope often increases (in absolute value) towards the edges while many real-world applications possess saturation characteristics.
- Many model architectures are (at least partly) based on polynomial approaches and therefore would be favored in an unfair manner.

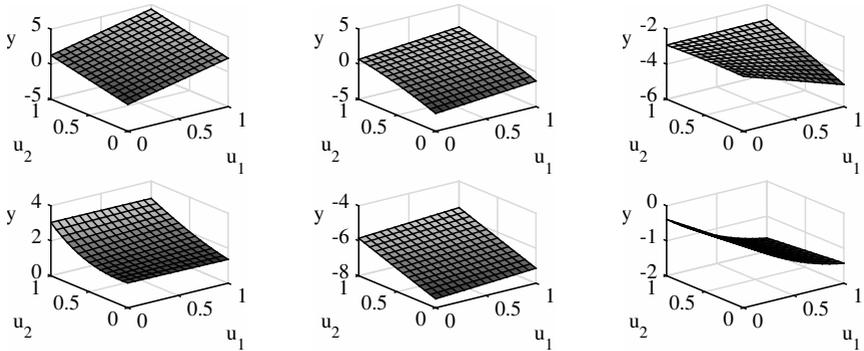


Figure 1: Six two-dimensional polynomial example functions ($M = 10$; $\mu = 1$).

In order to circumvent these drawbacks it is proposed to transform the polynomial with a sigmoid function:

$$h(u_1, u_2, \dots, u_n) = \frac{1}{1 + \exp(-\alpha \cdot g(u_1, u_2, \dots, u_n))}. \quad (5)$$

The tuning parameter α controls the amount of saturation. A recommended value that realizes a reasonable tradeoff between linearity and saturation is $\alpha = 10/\sqrt{M}$ since the variance of the polynomial output grows linearly with the number of terms M . Due to the normal distribution of the polynomial coefficients some polynomials will be hardly changed by the sigmoidal transformation since their output mainly utilizes the linear part of the sigmoid, while others will be strongly saturated, see Fig. 4. The output range of the strongly saturated polynomials can be very small and an appropriate scaling of the output might be necessary depending on the specific task or algorithm under investigation.

Comparison to Friedman's Approach

In 2001 Friedman introduced a random function generator to demonstrate the properties of gradient boosting [5]. This proposal addresses a general statistical setting. In contrast, the function generator discussed in this paper focuses on applications of engineering or similar type. In both cases application data might come from measurements or complex simulations (in the context of metamodeling). This section discusses the most relevant similarities and differences.

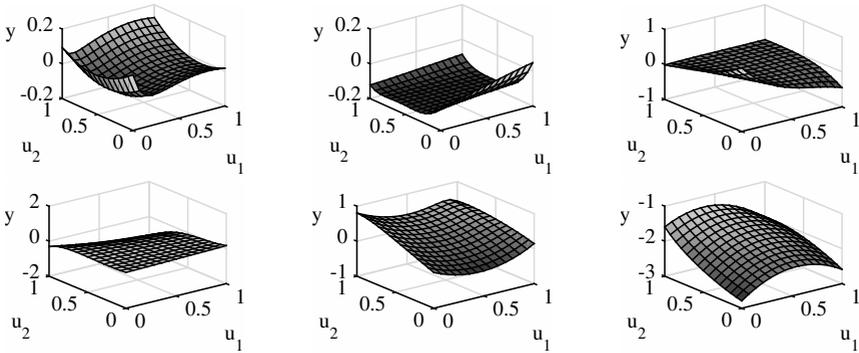


Figure 2: Six two-dimensional polynomial example functions ($M = 10$; $\mu = 3$).

Similarities

- Both function generators describe the output as a weighted superposition of nonlinear functions (additive structure). This paper proposes polynomial terms with random exponents, Friedman proposes Gaussian functions with full covariance matrices.
- The exponents (in our approach) and the standard deviations (i.e., square roots of the eigenvalues of the covariance matrices in Friedman's approach) are chosen randomly.
- The typical order of interactions is chosen much lower than the number of inputs. This paper accomplishes this effect by drawing the exponents from an exponential distribution (and taking the floor) which delivers many zero exponents implying that the corresponding inputs do not enter this polynomial term. Friedman's approach chooses the standard deviations from a uniform distribution and controls the interaction separately by selecting a random subset of the inputs for each Gaussian. The sizes of these subsets are also controlled via an exponential distribution (yielding an average value of 3.5).

Differences

- Data for engineering applications typically comes from active experimentation or relatively accurate simulations with mostly controllable input

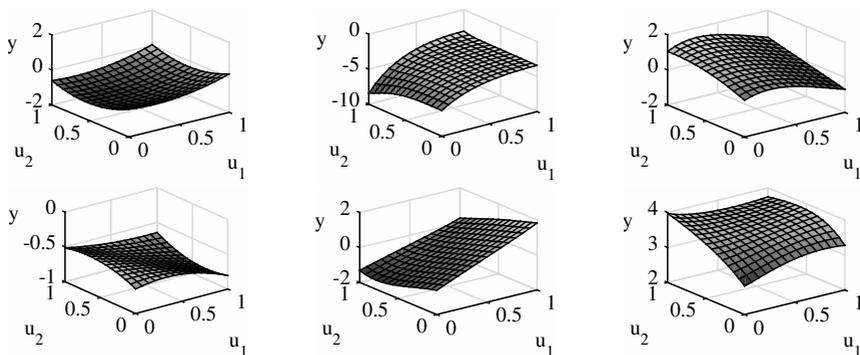


Figure 3: Six two-dimensional polynomial example functions ($M = 10$; $\mu = 2$).

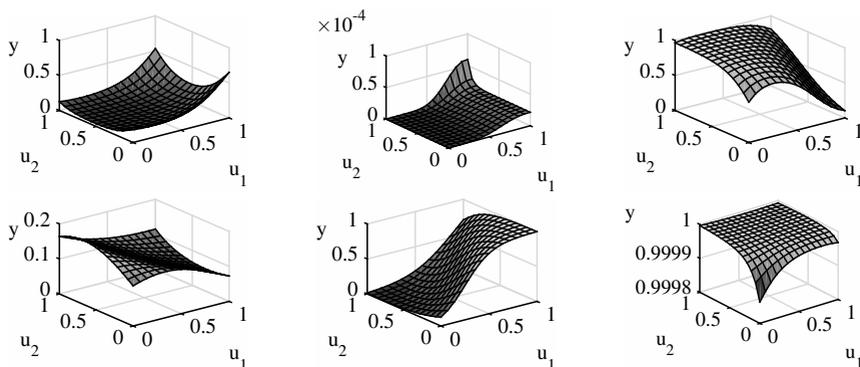


Figure 4: Six two-dimensional sigmoidally saturated polynomial example functions according to 5 ($M = 10$; $\mu = 2$).

variables while biological, economical, or social data often comes from passive observations with more non-controllable inputs.

Consequently, limiting the inputs between minimum and maximum values as proposed in this paper seems to be appropriate for modeling in engineering while input data drawn from a normal distribution as proposed in [5] tends to generate system behavior which typically is not found in engineering applications.

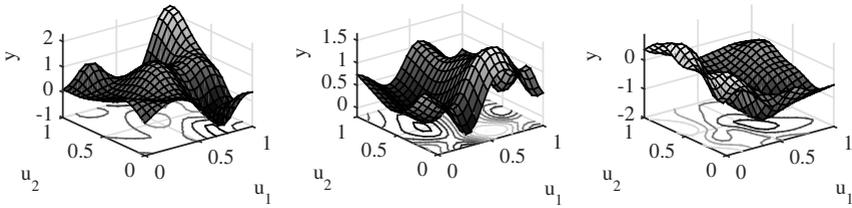


Figure 5: Three two-dimensional examples of Friedman's random functions with 20 Gaussians.

- The superposition of Gaussians in [5] likely yields non-monotonic or even multimodal relationships. Local minima and maxima are abundant, compare Fig. 5. This is not so commonly encountered in engineering applications. The approach proposed here is expected to exhibit this behavior less frequently. Therefore it seems to be more realistic in an engineering context.

Measuring Corners?

Typical human intuition favors the measurement of all inputs at their most extreme values, i.e., minima and maxima. In an unconstrained setting these combinations corresponds to the corners of the input space. Measuring these corners in n dimensions is called a 2^n full factorial design in DoE language [8]. Obviously, in a regression context additional measurements *within* the design space are absolutely necessary to detect any nonlinear behavior. Intuitive benefits of measuring corners are:

- Extrapolation can be completely avoided if the physical minima and maxima of each input are measured. This should tighten the confidence intervals and improve reliability.
- Arbitrarily high order interactions can be discovered in principle.
- Optima with respect to the inputs frequently occur at the boundary of the input space. Thus it might be a good idea to measure there in order to improve the accuracy of the model in these regions.

On the other hand, the number of corners (vertices) of an n -dimensional input space is 2^n . The following three cases can be distinguished (N = number of data points that shall be measured):

- Low-dimensional problems ($N \gg 2^n$): The number of corners is negligible to the overall number of points to be measured. Therefore it seems reasonable to include the corners to cover the most extreme combinations.
- Medium-dimensional problems ($N > 2^n$): This is the critical case mainly discussed in this paper and probably the most frequently occurring one (at least in an engineering context). It will be shown empirically that measuring corners offers increasing advantages when the amount of extrapolation and function complexity grow.
- High-dimensional problems ($N \leq 2^n$): The number of corners is similar or larger than the overall number of points to be measured. Therefore measuring corners is infeasible.

In principle the range between medium- and high-dimensional problems can be extended by relaxing from full factorial design 2^n (all corners are measured) to *fractional* factorial design 2^{n-m} where only $n - m$ inputs are measured in all combinations and the remaining m inputs are confounded. This can be seen as a compromise between measuring *all* corners and measuring *no* corners at all. This comes at the price of limiting the advantages with respect to extrapolation and high-order interactions.

Latin Hypercube Optimization with Corner Measurement

The simplest approach to enhance an LH design with corner points is to optimize a maximin LH as if no corner measurements were planned and afterwards just complement this design with the corner points. Thus if N design points shall be generated an $N - 2^n$ point maximin LH design is optimized and subsequently 2^n corner points are added. Figure 6a shows the result for $N = 10$ and $n = 2$ for a diagonal initialization of the LH optimization. For more details on the maximin LH optimization refer to [4]. Obviously, the overall design looks very inferior.

A much more sophisticated approach is to take the corner points into account during LH design optimization. Then the optimization automatically avoids to place LH points close to the corners and the first and last rows/columns of the LH design can be left blank because they are already taken many times by the corners. Figure 6b shows the much improved coverage yielded by this approach.

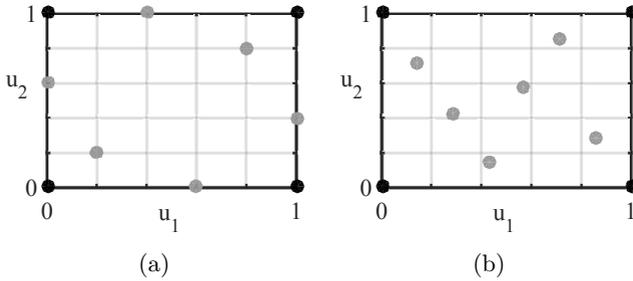


Figure 6: a) Optimal LH, afterwards complemented with corner points b) LH optimized taking the corners into account.

Note that the LH optimization proposed in [4] offers the substantial benefit of being able to include any number of prior design points. Therefore powerful multi-stage approaches with the incorporation of prior knowledge can easily be pursued.

However, one question remains: Is the explicit incorporation of corner points together with an LH design consisting of $N - 2^n$ points superior to an LH design with N data samples? To answer this question, the proposed function generator is utilized. Therefore 100 randomly generated functions for $\mu = 1$ and $\mu = 2$ for each input dimension $n = 2, \dots, 8$ are employed. The number of data samples is kept constant at $N = 300$, leading to a growing portion of corner points with an increasing input dimension. For each input dimension two LH designs (with N and $N - 2^n$ data samples) are optimized, which serve as training data for a local model network (LMN). The LMNs are trained using an incremental tree construction algorithm (HILOMOT = Hierarchical Local Model Tree) [10] that is capable to optimize an axes-oblique partitioning of the input space. The generalization performance of each LMN is evaluated on 100000 test points generated by a Sobol sequence [11]. All test data points lay in the hypercube $[0, 1]^n$. Three extrapolation scenarios are investigated:

- No extrapolation: Design space for training is $[0, 1]^n$.
- Small extrapolation: Design space for training is $[0.1, 0.9]^n$.
- Large extrapolation: Design space for training is $[0.2, 0.8]^n$.

Figure 7 shows the median test errors for complex saturated polynomials and all extrapolation scenarios. In case of no demanded extrapolation, the LH design without corners is superior for all input dimensions. The same holds

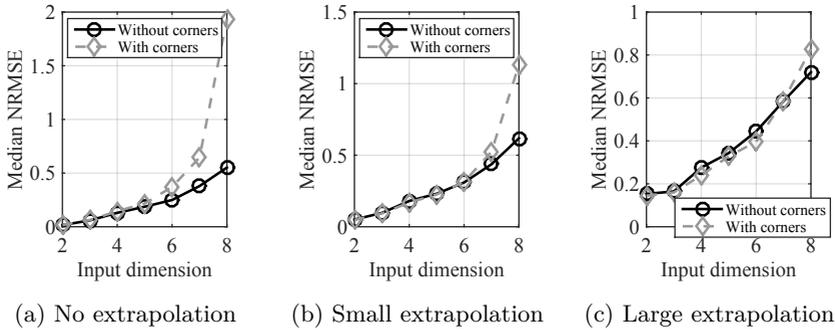


Figure 7: Normalized root mean squared error (NRMSE) on test data for complex ($\mu = 2$) saturated polynomials and all extrapolation scenarios.

for the small extrapolation case, but the benefit of the pure LH design is less pronounced. Looking at the median test errors of the large extrapolation scenario, the experimental design considering all corner points yields better generalization performance, except for the input dimensions seven and eight. The missing information from “inside” can not be overcompensated by more information at the boundaries. The results for other function complexities and polynomial functions (in contrast to saturated polynomial functions) are very similar, such that the findings hold for all investigated scenarios.

Application

This section discusses one example where the function generator proved to be a very useful tool. EU regulations (Ecodesign directive 2009/125/EC) set minimum efficiency requirements for fans sold on the EU markets and thereby creating an increased need for model-based aerodynamic optimization. Since optimization is not feasible using detailed computational fluid dynamics (CFD) simulations, these simulations are used instead to create a less complex metamodel. In the following, the terms “measurement” and “experiment” are used synonymously with “CFD simulation”. No physical measurements were carried out¹⁴. Nine critical geometric design parameters of the fan impeller such as the number of

¹⁴ For CFD simulation validation purposes a couple of fans will be physically built with rapid prototyping in the future.

blades, their geometric shape defined by several angles as well as the inner and outer diameter have been identified as metamodel inputs. Figure 8 shows a technical drawing in which almost all geometric design parameters are explained. The outputs are efficiency and pressure. For a more detailed discussion about fans refer to [2]. The fundamentals of fluid dynamics are covered in [9].

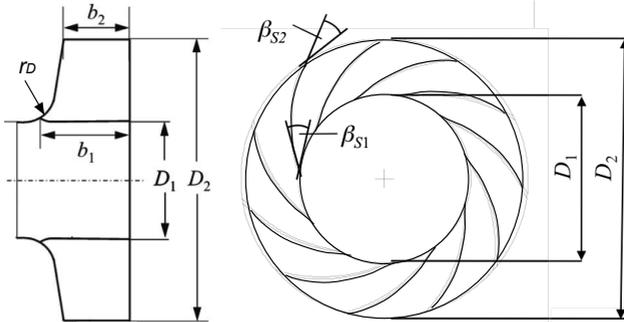


Figure 8: Geometric design parameters of a centrifugal fan

One crucial step in building such a metamodel is the design of experiments. Due to its advantages and simplicity a latin hypercube design [14] has been chosen (at least for a first phase of the project). In order to achieve good space filling properties the extended deterministic local search algorithm proposed in [4] has been utilized for all necessary LH designs throughout this project. For a current survey on the state of the art about optimization of maximin latin hypercubes refer to [13]. For a relatively recent discussion about the optimization of latin hypercubes refer to [6].

Setup

The question under investigation for the problem at hand, is the same as in the previous section: Should the corners of the design space be included in the experimental design and how does this influence the extrapolation behavior? Care is taken to avoid the necessity for extrapolation by choosing an appropriate range for the inputs. However, the risk of extrapolation can never be completely eliminated because the input range is chosen based on current expertise and therefore is due to change in the future.

The framework to address these questions is given by:

- Amount of data: About $N = 2000$ points can be generated by CFD simulations during the first phase of the project duration (~ 1 year).
- Dimensionality: $n = 9$.
- Strength of nonlinearity: Unknown.
- Data distribution: Suited to build a metamodel that can be utilized for industrial fan optimization. Optimization objectives may change. Therefore the input space should be covered in a space filling non-collapsing manner. All inputs are carefully chosen and highly relevant. Importance cannot be ranked a priori.
- Noise: Negligible. No measurement noise but small pseudo-random influences from the chosen CFD simulation settings like grid design.

Two experimental designs with 2048 points are compared:

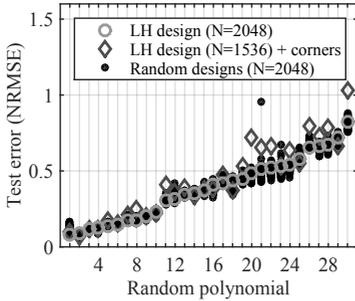
- Maximin latin hypercube with $N = 2048$ points.
- Maximin latin hypercube with $N = 1536$ points and $2^9 = 512$ corners of the 9D design space.

In addition to the two maximin latin hypercube designs, 50 random designs, where 2048 points are drawn from a uniform distribution are considered as well. The upper and lower bounds of the uniform distributions vary with the extrapolation scenarios described in the previous section. This ensures the same design space for the latin hypercube designs as well as for the random designs.

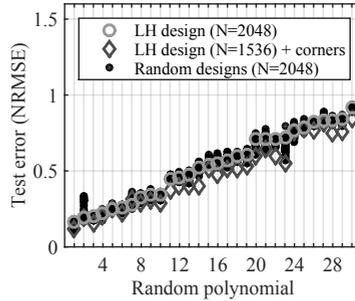
Simulations

In order to assess which quality can be expected, a metamodel is trained for 2048 data points with and without the corners, for 30 nonlinear functions randomly generated with the proposed function generator with $M = 30$ terms for simple ($\mu = 1$) and more complex ($\mu = 2$) functions. Polynomials (Poly) and saturated polynomials (SatPoly) are investigated.

As already described in the previous section, the HILOMOT training algorithm is used to train the LMNs. The input ranges and distributions for the train and test data sets are also adopted from the previous section, with the same extrapolation scenarios (no, small and large).



(a) Small extrapolation



(b) Large extrapolation

Figure 9: Normalized root mean squared errors (NRMSE) on test data for the complex polynomials ($\mu = 2$) for small and large extrapolation.

Results

As expected the inclusion of all corners in the experimental design becomes more important the more extrapolation is demanded. Representative for all investigations carried out, Fig. 9 shows the generalization performances for the complex random polynomials ($\mu = 2$) for small and large extrapolation scenarios. In case of no extrapolation the effect of the explicit consideration of all corners is always destructive for the model generalization performance. The shown results are sorted according to the generalization performance of the maximin latin hypercube design with $N = 2048$ points (without corners). Because the results for other function complexities and the sigmoidally saturated polynomials look very similar, we waive figures of these results.

The result is summarized in Table 1. The upper half represents the results for polynomial functions, the lower half for sigmoidally saturated polynomials. The first half is discussed first. In the case without extrapolation the corner points only harm the quality of the results. All 30 test functions can be approximated with a higher accuracy with all 2048 points placed on the maximin latin hypercube. The result is independent on the function's complexity. In contrast, in the large extrapolation case, the opposite is true: Data consisting of 1535 maximin latin hypercube points and 512 corner points yield superior results for all 30 test functions with $\mu = 1$ and 29 test functions with $\mu = 2$. The small extrapolation case lies in between. Whether corners should be measured depends on the complexity of the function.

Table 1: Measuring corners increases (+) or decreases (-) accuracy, number of functions where measuring corners is advantageous (out of 30) for polynomial and saturated polynomial functions.

Extrapolation Scenario	No	Small	Large
Poly Simple $\mu = 1$	-, 0	+, 24	+, 30
Poly Complex $\mu = 2$	-, 0	-, 9	+, 29
SatPoly Simple $\mu = 1$	-, 0	+, 21	+, 29
SatPoly Complex $\mu = 2$	-, 0	-, 11	+, 21

The second half of the table shows almost identical results for the saturated polynomials. Note that 30 *different*, randomly generated, polynomials have been used. However, the benefit that corner measurement can yield is reduced. Probably this is a consequence of simpler extrapolation behavior due to the saturation. This makes the benefits of corners less pronounced.

This result underlines the importance of being capable of systematically varying important properties of a function like its complexity. In the mentioned CFD simulation project it was decided to measure a latin hypercube without the corners since extrapolation was of minor importance. For other applications a reversed decision might be appropriate. The function generator allows to study these relationships individually for the specific situation at hand.

Conclusions

To be able to generate an arbitrary amount of test functions with specific properties, i.e. the dimensionality, the complexity as well as the amount of data, the proposed function generator is a very helpful tool for decision-making in the context of experimental modeling. It aims to imitate regression problems in engineering applications. Decisions during the development of algorithms or for the modeling task at hand can statistically be secured. We show, that with the help of our proposed function generator, we are able to choose an experimental design, that is most likely to yield good generalization performance for the metamodeling of a centrifugal fan for optimization purposes. We found that the

explicit consideration of corner points is very likely to harm the approximation quality of the metamodel since extrapolation is of minor importance. In addition to the application considered in this paper, we are able to give guidelines for the design of experiments in general, based on the expected extrapolation scenario and a process' complexity, see Table 1.

References

- [1] Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. MOA: Massive online analysis. *The Journal of Machine Learning Research*, 11:1601–1604, 2010.
- [2] Frank P Bleier. *Fan Handbook: selection, application, and design*. McGraw-Hill New York, 1998.
- [3] George EP Box, J Stuart Hunter, and William G Hunter. Statistics for experimenters: design, innovation, and discovery. *AMC*, 10:12, 2005.
- [4] Tobias Ebert, Torsten Fischer, Julian Belz, Tim Oliver Heinz, Geritt Kampmann, and Oliver Nelles. Extended deterministic local search algorithm for maximin latin hypercube designs. (accepted) In *Proc. IEEE Symposium on Computational Intelligence in Control and Automation*. IEEE, 2015.
- [5] Jerome H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.
- [6] Andrea Grosso, Armju Jamali, and Marco Locatelli. Finding maximin latin hypercube designs by iterated local search heuristics. *European Journal of Operational Research*, 197(2):541–547, 2009.
- [7] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*, volume 2. Springer, 2009.
- [8] Douglas C Montgomery. *Design and analysis of experiments*. John Wiley & Sons, 2008.
- [9] Bruce Roy Munson, Donald F Young, and Theodore Hisao Okiishi. *Fundamentals of fluid mechanics*. New York, 1990.
- [10] O. Nelles. Axes-oblique partitioning strategies for local model networks. In *IEEE International Symposium on Intelligent Control*, pages 2378–2383, Munich, Germany, Oct. 2006.

- [11] Harald Niederreiter. Low-discrepancy and low-dispersion sequences. *Journal of number theory*, 30(1):51–70, 1988.
- [12] Herschel Rabitz and Ömer F. Aliş. General foundations of high-dimensional model representations. *Journal of Mathematical Chemistry*, 25(2-3):197–233, 1999.
- [13] Arpad Rimmel, Fabien Teytaud, et al. A survey of meta-heuristics used for computing maximin latin hypercube. In Christian Blum and Gabriela Ochoa, editors, *Proceeding of the 14th European Conference on Evolutionary Computation in Combinatorial Optimisation*, pages 25–36, Granada, Spain, April 2014.
- [14] Thomas J. Santner, Brian J. Williams, and William Notz. *The design and analysis of computer experiments*. Springer, 2003.

Vergleich von Syntheseverfahren für Takagi-Sugeno Fuzzy Systeme anhand realer Raumklima-Daten

Alessio Cavaterra, Tarek Aissa, Steven Lambeck

Fachbereich Elektrotechnik und Informationstechnik, Hochschule Fulda
AG Präventives Klimamanagement
Leipziger Str. 123, 36039 Fulda
E-Mail: { alessio.cavaterra, tarek.aissa, steven.lambeck } @et.hs-fulda.de

Kurzfassung

Diese Arbeit beschäftigt mit sich einer nichtlinearen Systemidentifikation des Raumklimas. Einjährige Aufzeichnungen am konkreten Anwendungsfall Schloss Fasanerie bei Fulda liegen vor und bilden die Grundlage für die datengetriebene Modellbildung. Unzureichende Approximationsgüten von linearen Modellen der relativen Raumluftfeuchte motivieren zum Training von nichtlinearen Takagi-Sugeno Fuzzy Modellen. Hierbei werden gängige Verfahren zur Synthese dieser Modelle angewendet und hinsichtlich der Modellgüte, der Modellkomplexität, des Modellierungs- und Zeitaufwandes gegenüber gestellt. Die Untersuchungen beschäftigen sich insbesondere mit unterschiedlichen Prämissenstrukturen.

Einleitung

Diverse Forschungsarbeiten der AG Präventives Klimamanagement der HS Fulda zielen auf die Entwicklung von Regelstrategien für die relative Luftfeuchte in einem Raum mit Kunst- und Kulturgütern im Sinne der präventiven Konservierung ab. Schwankungen der relativen Luftfeuchte beschleunigen den Zerfallsprozess von Exponenten in Museen oder musealen Bauten [1]. Weil stationäre HVAC-Anlagen insbesondere für denkmalgeschützte Bauten nicht einsetzbar sind, wird hierbei auf mobile Be- und Entfeuchtungsgeräte zurück gegriffen. Mobile Geräte bieten einen Kostenvorteil, aber erfüllen die strengen Anforderungen der präventiven Konservierung bezüglich der relativen Luftfeuchte nicht. Die in den Geräten eingesetzten Zwei- oder Mehrpunktregler können im schlechtesten Falle eine unerwünschte

Schwankung der relativen Luftfeuchte begünstigen. Dies macht fortgeschrittene Regelstrategien, wie z.B. *Internal Model Control* oder *Modellbasierte Prädiktive Regelungen*, notwendig, die u.A. ein passendes Raumklimamodell benötigen.

Die Modelle in diesem Beitrag beruhen hauptsächlich auf dem ARX-Modellansatz (*autoregressive with exogenous input*) der Form

$$\hat{y}(k) = \sum_{i=1}^{n_b} b_i \cdot u(k-i) - \sum_{i=1}^{n_a} a_i \cdot y(k-i)$$

$$\hat{y}(k) = {}^k \hat{y}(\mathbf{x}) = \mathbf{\Gamma} \mathbf{x} = (b_1, \dots, b_{n_b}, -a_1, \dots, -a_{n_a}) \begin{pmatrix} u(k-1) \\ \vdots \\ u(k-n_b) \\ y(k-1) \\ \vdots \\ y(k-n_a) \end{pmatrix}. \quad (1)$$

Die Eingänge $u(k-1) \dots u(k-n_b)$ und Ausgänge $y(k-1) \dots y(k-n_a)$ des Modells werden mit den Wichtungen $b_1 \dots b_{n_b}$, $a_1 \dots a_{n_a}$ multipliziert. k beschreibt in Kurzform den Abtastschritt $k = kT_A$. Die genannten Wichtungsparameter können mit Hilfe des klassischen Kleinste-Quadrate-Schätzers bestimmt werden.

Der ARX-Modellansatz in Gleichung (1) wird des Weiteren für nichtlineare Takagi-Sugeno Fuzzy Modelle (TS) verwendet [2]. Die synthetisierten Modelle werden in Kurzform mit *TS+Kürzel des Verfahrens* bezeichnet (z.B. TSLOL für ein TS-Modell in Verbindung mit dem LOLIMOT-Verfahren). Der Schedulingvektor \mathbf{z} des TS-Eingangsraumes ist dem Regressorenvektor \mathbf{x} in diesem Fall gleich. Die einzelnen Ausgänge der ARX-Teilmodelle $\hat{y}_i(k) = {}^k \hat{y}_i(\mathbf{x})$ werden über die Fuzzy-Basisfunktionen $\phi_i(\mathbf{x})$ gewichtet. Der gesamte Ausgang $y(\mathbf{x})$ eines TS-Modells kann im vorliegenden Fall als die Summe aller gewichteten c ARX-Teilmodelle (Regeln) verstanden werden.

$$y(\mathbf{x}) = \frac{\sum_{i=1}^c \mu_i(\mathbf{x}) \cdot {}^k \hat{y}_i(\mathbf{x})}{\sum_{i=1}^c \mu_i(\mathbf{x})} = \sum_{i=1}^c \phi_i(\mathbf{x}) \cdot {}^k \hat{y}_i(\mathbf{x}) = \sum_{i=1}^c \phi_i(\mathbf{x}) \cdot \mathbf{\Gamma}_i \mathbf{x}. \quad (2)$$

Die Fuzzy-Basisfunktionen $\phi_i(\mathbf{x})$ haben ihren Wertebereich zwischen Null und Eins. Berechnet werden sie aus dem Mittel der Zugehörigkeitsgrade

$$\phi_i(\mathbf{x}) = \frac{\mu_i(\mathbf{x})}{\sum_{i=1}^c \mu_i(\mathbf{x})}, \quad (3)$$

was die Orthogonalität der Basisfunktionen sicherstellt $\sum_{i=1}^c \phi_i(\mathbf{x}) = 1$.

Alle $\mu_i(\mathbf{x})$ Zugehörigkeitsfunktionen (ZF) werden im folgenden als Gaußglocken festgelegt; deren Funktion lautet mit dem Zentrum ν und der Standardabweichung σ :

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

Für die Schätzung der Parameter Γ_i im TS-Modell nach Gleichung (2) wird nun ein gewichteter Kleinste-Quadrate-Ansatz (WLS, *weighted-least-squares*) verwendet, dessen Gewichtung eine Diagonalmatrix aus den Fuzzy-Basisfunktionen $\phi_i(\mathbf{x})$ darstellt. Dies unterstützt die Schätzung der einzelnen Teilmodelle.

Die anschließenden Untersuchungen verwenden Fehler- bzw. Gütemaße wie den RMSE (*root-mean-squared-error*) und den VAF (*variance-accounting-for*) [3]. Zuletzt genanntes Gütemaß wird ausschließlich im Abschnitt „Clustering“ verwendet. Die Einheit des RMSE ist je nach Temp.- oder Feuchtemodell entweder in °C oder in %rH festgelegt.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (5)$$

$$\text{VAF} = 100\% \cdot \left(1 - \frac{\text{Var}(y_i - \hat{y}_i)}{\text{Var}(y_i)}\right) \quad (6)$$

Die Maße in den vorhergehenden Gleichungen benötigen die Modellausgänge \hat{y}_i , $i = 1 \dots N$, welche hierbei in nachfolgenden Betrachtungen über eine vollständige Simulation gewonnen werden. Die Simulation der einfachen ARX-Modelle in (1) lässt die Parameter nichtlinear in deren Ausgang eingehen, sodass eine (W)LS-Schätzung inkonsistent ist. Korrekturverfahren werden in [4] vorgeschlagen.

TS-Modell mit Gitter-Partitionierung (TSGIT)

Der Eingangsraum eines TS-Modells beinhaltet alle Variablen, die zur Auswertung der Regelbasis gebraucht werden. Im übertragenen Sinne wird nach diesen Einflussgrößen *geplant*, sodass diese auch als *Schedulingvariablen* bezeichnet werden. Wie im Vorfeld erwähnt, sind die Schedulingvariablen und die Regressoren im Konklusionsteil der TS-Modelle identisch gewählt. Die Schedulingvariablen (bzw. die Regressoren) der Gleichung (7) legen demnach einen 4-dimensionalen Eingangsraum fest. Diese sind zuvor durch eine modellfreie Strukturidentifikation ermittelt worden [5].

$$\mathbf{x} = (\varphi_i(k-1), \vartheta_i(k-1), \varphi_a(k-1), \vartheta_a(k-1))^T \quad (7)$$

Jede Größe wird nun mit zwei ZF aufgelöst. Die Parameter der ZF, in diesem Fall die Zentren ν_i und die Standardabweichungen σ_i , werden im Vorfeld festgelegt. Hierdurch entsteht ein reguläres oder irreguläres Gitter im Eingangsraum, welches mehr oder weniger einem Schachbrett ähnelt [6]. Nach Festlegung der Prämisse

Tabelle 1: Parameter der Gauß'schen-ZF von TSGIT. Einheitenlos.

Sched.-var.	ν	σ
$\varphi_i(k-1)$	33, 66	8, 8
$\vartheta_i(k-1)$	8, 16	2, 2
$\varphi_a(k-1)$	33, 66	8, 8
$\vartheta_i(k-1)$	8, 16	2, 2

(vgl. Tab. 1 und Bild 1) werden die Parameter des Konklusionsteils mit einem WLS-Schätzer identifiziert. Die gitterförmige Partitionierung macht eine vollständig verknüpfte Regelbasis notwendig, die im vorliegendem Fall $2^4 = 16$ unterlagerte Teilmodelle voraussetzt. Insgesamt sind $4 \times 16 = 64$ Wichtungsparameter zu schätzen.

Die Schätzung des Konklusionsteils erfordert höheren Rechenaufwand, da 16-mal eine WLS-Schätzung stattfindet. Zeitlich fällt diese allerdings deutlich weniger ins Gewicht als die Simulation des Modells. Die Ergebnisse zeigen (s. Bilder 2 und 3), dass mit TSGIT die Modellgüte um circa 20 Prozent verbessert werden konnte, als mit einem linearen Ansatz möglich ist. TSGIT erzielt auf dem Trainingsdatensatz $RMSE_{tr} = 3.14\%rH$ und auf dem Testdatensatz $RMSE_t = 3.18\%rH$. Dynamische Signalanteile werden im Vergleich zu einem einzelnen ARX-Modell besser abgebildet.

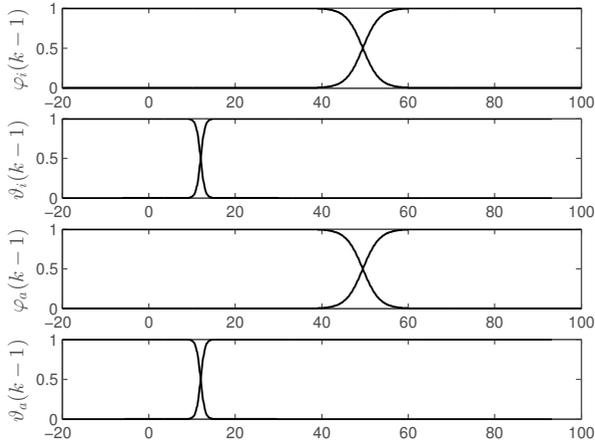


Bild 1: Basisfunktionen im Eingangsraum von TSGIT.

Optimierung des Eingangsraumes von TSGIT (TSOPT)

Dieser Unterabschnitt behandelt die Frage, ob optimierte Zentren und Standardabweichungen der ZF zu höheren Modellgüten führen können. Der Eingangsraum des TSGIT-Modells wird dazu beibehalten. Lediglich die Parametrierungen der ZF werden mit Hilfe der Differentialevolution [7] optimiert. Das Resultat ist ein optimiertes TSGIT-Modell, was im Folgenden mit TSOPT bezeichnet wird.

Der umfangreiche Datensatz erhöht den Rechenaufwand bei der Optimierung. Um diesen Nebeneffekt einzudämmen, wird ein zweiwöchiger Trainingsdatensatz und ein einwöchiger Testdatensatz (beide zeitlich zusammenhängend) aus den Daten ausgewählt. Diese dienen während des Optimierungsvorganges als Datengrundlage. Ist das Optimierungsverfahren abgeschlossen, stellt TSOPT das optimale Modell für die beiden kleinen Datensätze dar. Da die Struktur des Eingangsraumes und weniger die Parameter der Konklusionen wichtig sind, wird die Struktur von TSOPT auf den ursprünglichen Trainings- und Testdatensätzen trainiert bzw. getestet. Dieser Schritt verändert den Eingangsraum von TSOPT nicht, aber besetzt die Konklusionsparameter mit neuen Werten. Hierdurch wird ein Vergleich mit TSGIT (und anderen TS-Modellen) möglich.

Wie jedes Optimierungsverfahren benötigt die Differentialevolution eine Initialisierung der eigenen Parameter. Der Mutationskalar F und die Auswahlwahrscheinlichkeit w werden beide zu $F = w = 0.9$ festgelegt (vgl. [7]). Die Population be-

steht aus insgesamt 32 Individuen. Ein Individuum ist ein TS-Modellkandidat mit 16 Optimierungsvariablen (8 ZF \times 2 ZF-Param. / Sched.-var.). Nach $g_{\max} = 400$ Generationen soll der Optimierungsvorgang abgebrochen und das beste TS-Modell ausgegeben werden.

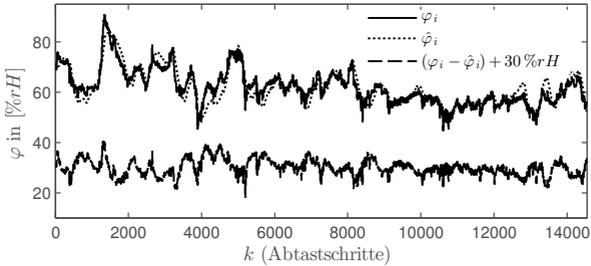


Bild 2: Simulation von $\hat{\varphi}_i$ mit TSGIT auf dem Trainingsdatensatz.

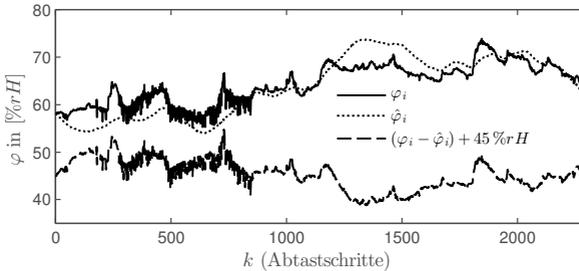


Bild 3: Simulation von $\hat{\varphi}_i$ mit TSGIT auf dem Testdatensatz. Aus Darstellungsgründen wird der absolute Fehler mit einem Offset beaufschlagt.

Eine sehr wichtige Aufgabe ist die Bewertung der Fitness eines einzelnen Individuums. Die Fitness wird über die Berechnung eines zuvor festgelegten Kostenfunktionalen bestimmt. Neben der Modellgüte auf dem (verkleinertem) Trainings- und dem Testdatensatz müssen weitere Bedingungen berücksichtigt werden, die von den RMSE-Gütemaßen nicht abgedeckt werden. Verletzt ein Individuum solch eine Bedingung, tritt eine Strafe in Kraft, die die Fitness des Individuums

verringert. Das Kostenfunktional setzt sich also aus den beiden RMSE-Gütemaßen und einer Strafe J_s zusammen.

$$J = \text{RMSE}_{tr} + \text{RMSE}_t + J_s \quad (8)$$

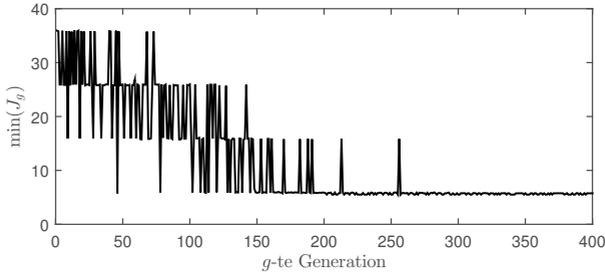


Bild 4: Entwicklung des minimalen Kostenwertes $\min(J_g)$.

Der Strafterm J_s nimmt alle gewünschten Einschränkungen in Form von Teilstrafen J_i auf, die im Folgendem allesamt mit ganzzahligen Werten festgelegt werden.

$$J_s = \sum_{i=1}^4 J_i \quad \text{mit} \quad (9)$$

$$J_1 = \begin{cases} 10 & \text{wenn } \nu_a < \nu_b \quad \text{mit } a < b, (a, b) \in \mathbb{N} \\ 0 & \text{sonst.} \end{cases}, \quad (10)$$

$$J_2 = \begin{cases} 10 & \text{wenn } \sigma_a < 0 \quad \text{mit } a \in \mathbb{N} \\ 0 & \text{sonst.} \end{cases}, \quad (11)$$

$$J_3 = \begin{cases} 10 & \text{wenn } \nu_a < \epsilon_u \quad \text{mit } a \in \mathbb{N} \\ 10 & \text{wenn } \nu_a > \epsilon_o \\ 0 & \text{sonst.} \end{cases}, \quad (12)$$

$$J_4 = \begin{cases} 1000 & \text{wenn } \hat{y}_i = \text{NaN} \quad \text{mit } i \in \mathbb{N} \\ 0 & \text{sonst.} \end{cases}. \quad (13)$$

Strafe (10) bestraft eine Vertauschung von ZF, die späteren Interpretationen im Wege steht. Die Strafen (11) und (12) verkleinern den Suchraum. Es existieren keine negativen Standardabweichungen und Zentren außerhalb des Wertebereiches

sind unerwünscht. Fehlgeschlagene Parameterschätzungen bzw. Simulationen sind durch NaN (*not-a-number*) gekennzeichnet und führen zu inakzeptablen Modellgüten. Die Teilstrafen J_i werden allesamt in einfachen Wenn-Dann-Strukturen abgefragt und anschließend in J_s mit der Beziehung (9) aufsummiert.

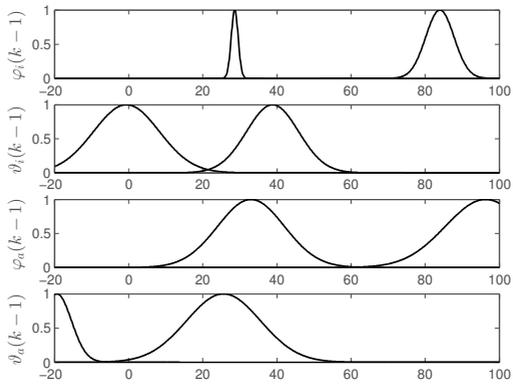
Tabelle 2: Eingangsraum von TSOPT. Auf Einheiten wird verzichtet.

Sched.-var.	ν	σ
$\varphi_i(k-1)$	28.6, 84.0	0.93, 3.81
$\vartheta_i(k-1)$	-0.6, 38.8	8.88, 7.02
$\varphi_a(k-1)$	33.0, 96.3	8.98, 10.83
$\vartheta_i(k-1)$	-19.3, 25.6	3.93, 9.97

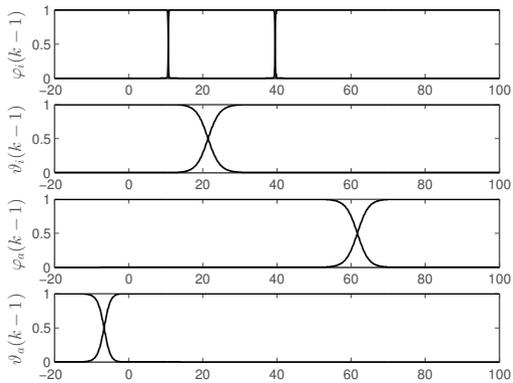
Bild 4 stellt die geringsten Kosten der Population in der g -ten Generation $\min(J_g)$ dar. Bereits nach $g = 200$ Generationen haben die Individuen offensichtlich eine optimale Konfiguration des Eingangsraumes auf den angepassten Datensätzen angenommen. Das beste Individuum stellt das TSOPT-Modell mit $J = 5.51$ dar. Die Parameterwerte des Eingangsraumes sind in der Tabelle 2 aufgeführt (vgl. auch Bild 5).

Wie im Vorfeld angekündigt, werden die Parameter der Teilmodelle von TSOPT erneut auf dem ursprünglichen Trainingsdatensatz geschätzt. Anschließend wird das Modell auf dem Testdatensatz simuliert. Die Resultate sind in der Bild 6 abgebildet. TSOPT erreicht Gütemaße von $\text{RMSE}_{tr} = 3.29\%rH$ und $\text{RMSE}_t = 2.29\%rH$. TSOPT verschlechtert sich marginal gegenüber TSGIT auf dem Trainingsdatensatz, aber schneidet auf dem Testdatensatz gut ab.

In Bild 5b ist ein Reaktivierungseffekt der Basisfunktion $\phi_2(\varphi_i(k-1))$ im Eingangsraum der relativen Raumluftfeuchte zu erkennen. Dieser entsteht bei Vorhandensein von ZF mit unterschiedlichen σ . Die oben links in Bild 5a dargestellte ZF $\mu_1(\varphi_i(k-1))$ ist für $\varphi_i(k-1) \lesssim 20\%rH$ im Betrage kleiner als $\mu_2(\varphi_i(k-1))$ für den gleichen Wertebereich. Hierdurch reaktiviert sich die Basisfunktion der zweiten ZF. Solche Effekte können bei Identifikationsaufgaben zu Problemen führen. Untersuchungen und Gegenmaßnahmen sind in [4] zu finden.

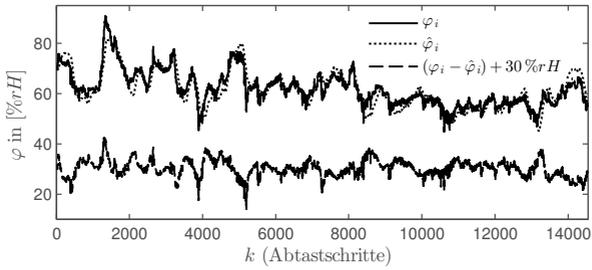


(a) Zugehörigkeitsfunktionen.

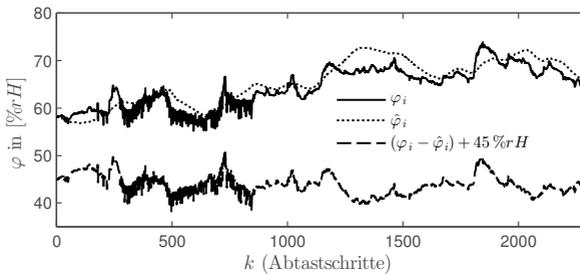


(b) Basisfunktionen.

Bild 5: Eingangsraum von TSOPT.



(a)



(b)

Bild 6: Simulation von $\hat{\varphi}_i$ mit TSOPT. a Trainings- und b Testdatensatz. Aus Darstellungsgründen wird der absolute Fehler mit einem Offset beaufschlagt.

Synthese mittels Clusteranalyse (TSGK)

Clusterverfahren sind ein Teilgebiet der Mustererkennung und beschreiben die Suche nach zusammenhängenden Datengruppen (Klassen). Unschärfe Clusterverfahren klassifizieren Daten in einer Art und Weise, welche keine 100 %-ige Klassenzuordnung zwingend voraussetzen. Ein Datenpunkt kann demnach zwei oder mehr Klassen zu gewissen Graden zugehörig sein. Ein etabliertes Fuzzy-Clusterverfahren ist der Gustafson-Kessel-Algorithmus (GK), der ellipsoide Cluster sucht [6]. Er eignet sich insbesondere zur Identifikation von TS-Modellen, da die Eingangs- und Ausgangsdaten nach zusammenhängenden Daten gruppiert und hieraus direkt ZF aus den geometrischen Eigenschaften extrahiert werden können (Produktraum-Clustering). Die ZF erfüllen zudem die Orthogonalitätsbedingung $\sum_i \mu_i = 1$.

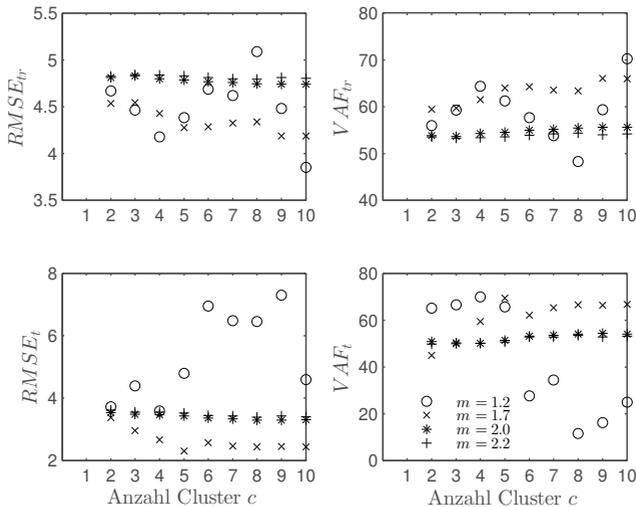


Bild 7: RMSE u. VAF (jeweils gemittelt aus 10 Modellen) über Clusteranzahl c .

Der GK benötigt Initialparameter. Die wichtigsten stellen die Anzahl der Cluster c und die sogenannte *Fuzziness* $m > 1$ dar. Letzterer kann als Unschärfeparameter interpretiert werden, der die Härte zwischen den Übergängen der Cluster beeinflusst. Ein großes m bewirkt eine hohe Unschärfe der ZF. Bei einer Black-Box-Modellierung fällt eine gute Wahl von c und m oftmals schwierig. Vorschläge sind hierzu in [6] zu finden. Im Folgenden wird die Wahl dieser Parameter anhand

der Fehler- bzw. Gütemaße RMSE und VAF getroffen. Bild 7 stellt die gemittelten Maßzahlen über der Clusteranzahl dar. Für jedes c sind 10 Modelle trainiert und getestet worden. Weiterhin ist der Einfluss der Fuzziness $m = \{1.2, 1.7, 2.0, 2.2\}$ aufgeführt. Anhand der Resultate auf dem Testdatensatz werden $c = 5$ Cluster mit einer Fuzziness von $m = 1.7$ bestimmt. Eine hohe Güte des TSGK-Modells ist also für die genannte Konfiguration zu erwarten.

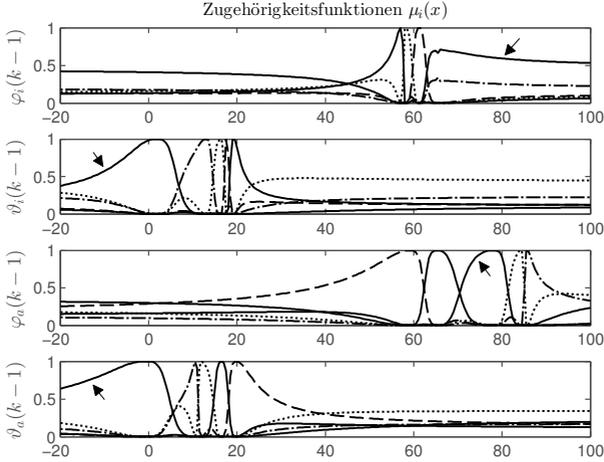


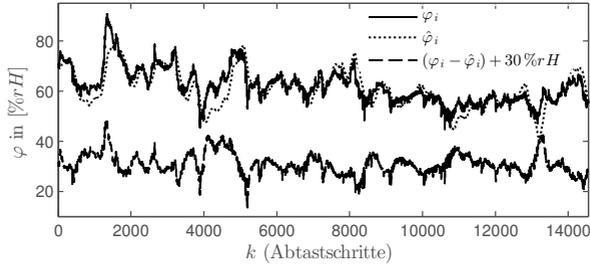
Bild 8: Mit dem GK-Algorithmus geclusterter Eingangsraum von TSGK.

Der partitionierte Eingangsraum ist in dem Bild 8 dargestellt. Die eingetragenen Pfeile heben ZF hervor, die ein Cluster bzw. ein Modell beschreiben. Die Extraktion der ZF geschieht mittels GK sehr einfach. Über die Vektoren der Clusterzentren $\mathbf{v}_1, \dots, \mathbf{v}_c$, dem Datenpunktvektor \mathbf{z} , den Formmatrizen $\mathbf{A}_1, \dots, \mathbf{A}_c$ und dem Distanzmaß

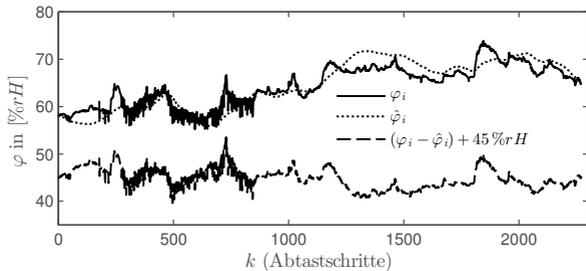
$$d^2(\mathbf{z}, \mathbf{v}_i) = (\mathbf{z} - \mathbf{v}_i)^T \mathbf{A}_i (\mathbf{z} - \mathbf{v}_i),$$

lassen sich die ZF $\mu_i(\mathbf{z})$ nach der probabilistischen Methode berechnen. Die extrahierten ZF weisen für $\mu_i = 1$ auf die vollständige Zugehörigkeit eines Datenpunktes zu einem Clusterzentrum hin.

$$\mu_i(\mathbf{z}) = \frac{1}{\sum_{j=1}^c \left(\frac{d^2(\mathbf{z}, \mathbf{v}_i)}{d^2(\mathbf{z}, \mathbf{v}_j)} \right)^{\frac{1}{m-1}}}. \quad (14)$$



(a)



(b)

Bild 9: Simulation von $\hat{\varphi}_i$ mit TSGK. a Trainings- und b Testdatensatz. Aus Darstellungsgründen wird der absolute Fehler mit einem Offset beaufschlagt.

Die Simulation von TSGK auf dem Trainingsdatensatz liefert ein Fehlermaß $RMSE_{tr} = 4.30\%rH$, der aufgrund der dürtig abgedeckten Dynamik im ersten Teil der Trainingsdaten entsteht. Auf dem Testdatensatz wird $RMSE_t = 2.15\%rH$ erreicht. Auch hier sind Schwächen bei der Dynamik ersichtlich. Allerdings erbringt hier das TSGK-Modell den bisher niedrigsten RMSE-Wert (vgl. Bild 9). Die Synthese und Simulation von TSGK ist mit Hilfe der MATLAB[®] Fuzzy Identification Toolbox¹⁵ durchgeführt worden.

¹⁵ Version 4.0, <http://www.dsc.tudelft.nl/~rbabuska/>

Synthese mit LOLIMOT-Algorithmus (TSLOL)

In einem vorhergehendem Abschnitt hat der Anwender durch Wahl der Anzahl und der Parametrierung der ZF direkt in die Prämissenstruktur eingegriffen und so mehr oder weniger reguläre Gitterpartitionen erstellt. Eine systematische Konstruktion mit ähnlichem Prinzip erscheint sinnvoll. LOLIMOT (*local-linear-model-tree*) ist ein Konstruktionsalgorithmus, der den Eingangsraum dort achsorthogonal aufteilt (halbiert), wo Partitionen eine schlechte Modellgüte aufweisen. In jeder Iteration wird so eine neue Regel zum Modell hinzugefügt (irreguläres Gitter), sodass LOLIMOT zu den Algorithmen mit einer Wachstumsstrategie gezählt wird. Üblicherweise werden Gauß'sche-ZF verwendet. Im Gegensatz zu anderen Verfahren benötigt LOLIMOT nur einen Einstellparameter $k_\sigma = \frac{\sigma_{ij}}{\Delta_{ij}}$, der das Verhältnis zwischen den Standardabweichungen und der Kantenlänge der Hyperrechtecke bildet. Im Folgenden sei $k_\sigma = 1/3$ gewählt, da sich dieser Wert oftmals als gute Wahl erweist. Umfassende Untersuchungen zu LOLIMOT sind in [4] aufgeführt.

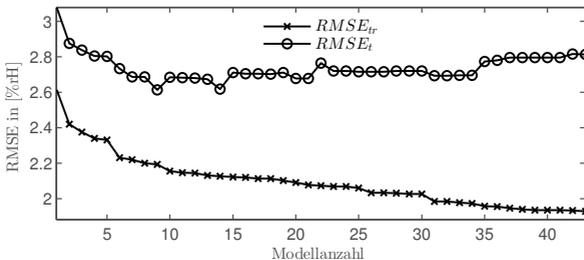


Bild 10: Beste Prämissenstruktur für $c = 9$ Modelle.

Ähnlich wie beim Produktraum-Clustering stellt sich die Frage nach einer guten Modell- bzw. Regelanzahl, wenn der Datensatz mit Rausch- bzw. mit Störsignalen behaftet ist. Die Suche nach einer optimalen Modellanzahl stellt sich hierbei als sehr einfach heraus, da der Rechenaufwand für LOLIMOT vergleichsweise gering ist. Das Bild 10 stellt den Verlauf der genannten Fehlermaße während der Anwendung von LOLIMOT dar. Das Minimum auf dem Testdatensatz empfiehlt die Verwendung von $c = 9$ Modellen.

Der Eingangsraum in Bild 11 zeigt zunächst seltsam wirkende ZF, die sich über andere ZF aufspannen. Diese entstehen durch die achsorthogonale Halbierung der Partitionen, die ein irreguläres Gitter hervorrufen. Solche ZF können linguistisch als *don't care* interpretiert werden. LOLIMOT führt demnach eine Optimierung

der Prämissenstruktur bzw. der Regelbasis durch. Der Eingangsraum von ϑ_a ist ausschließlich mit *don't care*-ZF belegt, sodass LOLIMOT hier keinerlei Halbierung durchgeführt hat. Eine Verwendung von ϑ_i als Schedulingparameter ist demnach in TSLOL als überflüssig zu interpretieren.

TSLOL erreicht Fehlermaße auf dem Trainings- und Testdatensatz von $RMSE_{tr} = 2.19\%rH$ und $RMSE_t = 2.61\%rH$. Das etwas schlechtere Ergebnis auf dem Testdatensatz ergibt sich aufgrund eines im Trainingsdatensatz spärlichen besetzten Wertebereichs $4^\circ C \dots 10^\circ C$ der Raumtemperatur ϑ_i . LOLIMOT tendiert dazu dünn besiedelte Datenbereiche weniger aufzuteilen [4]. Dieser Effekt ist an ϑ_i in Bild 11 einsehbar.

Die in diesem Abschnitt besprochene Modellierung des Raumklimas hat mit Hilfe der MATLAB[®]-LMNToolbox¹⁶ stattgefunden. Weitere Informationen sind in [8] aufzufinden.

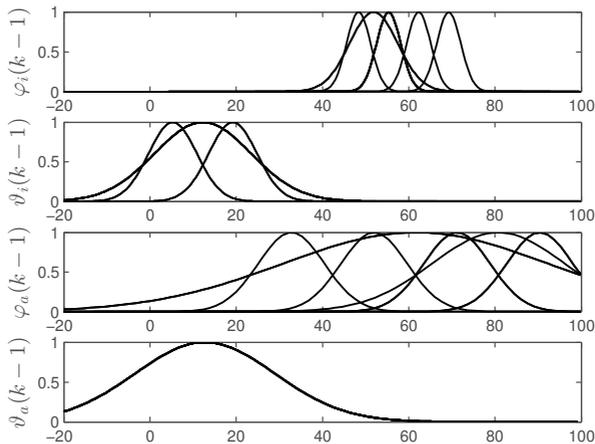
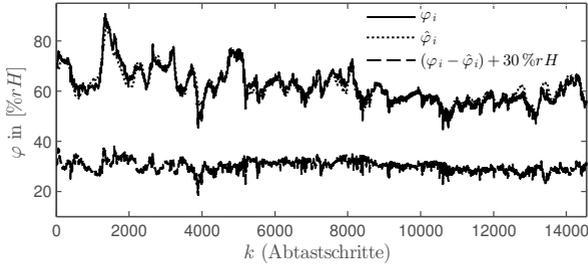


Bild 11: Irreguläre Gitterpartitionierung durch LOLIMOT.

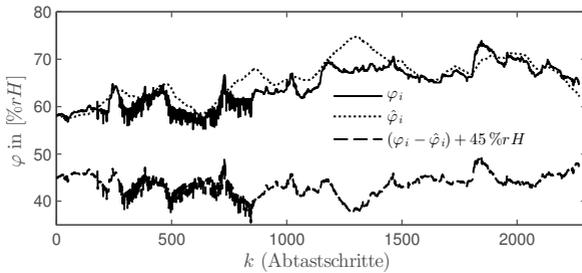
Vergleich der Ansätze

Während ein einfaches Modell der Temperatur zu zufriedenstellenden Ergebnissen führt, müssen, aufgrund der starken Nichtlinearität zwischen der relativen

¹⁶ <http://www.mb.uni-siegen.de/mrt/lmn-tool/>



(a)



(b)

Bild 12: Simulation von $\hat{\varphi}_i$ mit TSOPT. a Trainings- und b Testdatensatz. Aus Darstellungsgründen wird der absolute Fehler mit einem Offset beaufschlagt.

Luftfeuchte und der Temperatur, einfache ARX-Ansätze zur Modellierung des Feuchteverhaltens verworfen werden. Die Nutzung von TS-Modellen zeigt bei bereits sehr einfachen gitterförmigen Syntheseansätzen (TSGIT) Erfolg und motiviert zu weiteren Verbesserungen der Prämissenstruktur. Das TSOPT-Modell baut auf TSGIT auf, indem die Prämissenstruktur mitsamt vollständigem Regelwerk übernommen, aber die Lage und Breite der Gauß'schen-ZF mit einem Evolutionärem Algorithmus optimiert werden. Die Optimierung bewirkt eine höhere Modellgüte als TSGIT, aber benötigt durch das Optimierungsverfahren viel Rechenzeit und verursacht großen Rechenaufwand. Hinzu kommt, dass eine sauber formulierte Kostenfunktion zur Berücksichtigung von Nebeneffekten, wie z.B. dem Reaktivierungseffekt, notwendig ist. Beide gitterförmig partitionierten Modelle lassen jedoch eine gute Interpretation der Prämisse durch linguistische Ausdrücke zu.

Tabelle 3: Vergleich der TS-Modelle. (++) sehr vorteilhaft, (+) vorteilhaft, (0) akzeptabel, (-) wenig vorteilhaft, (--) unvorteilhaft.

Kriterium	TSGIT	TSOPT	TSGK	TSLOL
RMSE _{tr} in %rH	3,14	3,29	4,30	2,19
RMSE _t in %rH	3,18	2,29	2,15	2,61
Modell- bzw. Regelanz.	16	16	5	9
Anz. Konklusionsparam.	64	64	20	36
Interpretation	++	+	0	++
Rechenzeit	++	--	+	++
Initialparam.	-	--	+	+

Durch eine Clusteranalyse beziehungsweise ein Produktraum-Clustering wird das TSGK-Modell identifiziert, welches im Vergleich zu den vorhergehenden Modellen eine deutlich geringere Regelbasis (Modellkomplexität) besitzt und auf dem Testdatensatz am besten abschneidet. Die extrahierten ZF sind im Gegensatz zu den vorhergehenden Modellen nur schwer zu interpretieren. Die Wahl guter Clustering-Parameter ist durch eine vorhergehende heuristische Suche ermöglicht worden, die wiederum mehr Rechenaufwand beansprucht.

Das Syntheseverfahren mit Hilfe des LOLIMOT-Algorithmus ermöglicht eine linguistische Interpretation der Prämisse, nutzt die Vorteile einer irregulären Gitterpartitionierung und liefert im Vergleich zu vorhergehenden Ansätzen sehr schnell ein Modell. Das synthetisierte TSLOL zeigt hohe Approximationseigenschaften auf dem Trainings- und dem Testdatensatz und empfiehlt die Reduktion des Schedulingvektors um die Raumtemperatur. Die Modellgüte auf dem Testdatensatz ist etwas schlechter als bei TSGK, was auf spärlich besetzte Datenbereiche zurückzuführen ist, da LOLIMOT die Aufteilung solcher meidet.

Zusammenfassung

Die Untersuchungen in dieser Arbeit haben sich mit verschiedenen Syntheseverfahren von Takagi-Sugeno Fuzzy Modellen zur Identifikation des Raumklimas anhand realer Messdaten beschäftigt. Der Schwerpunkt lag hierbei in der Wahl der Prämissenstruktur eines TS-Modells, die durch die Anzahl und die Form der ZF sowie der Regelbasis festgelegt wird. Da die Modellierung der relativen Raumluftfeuchte mit einfachen ARX-Ansätzen zu wenig zufriedenstellenden Resultaten

fürten, wurden als nächstes TS-Modelle mit vollständigen Regelwerken untersucht, die eine gitterförmige Partionierung des Eingangsraumes vorsehen (TSGIT und TSOPT). Das TSOPT baut auf der Prämissenstruktur von TSGIT auf und ist diesbezüglich optimiert worden. Zur Reduktion der Regelbasis wurden automatisierte Verfahren wie das Produktraum-Clustering (TSGK) oder LOLIMOT (TSLOL) eingesetzt, die besonders schnelle und gute Modellierungsergebnisse lieferten.

Weiterführende Arbeiten sehen die Reduktion des Schedulingvektors und die erneute Suche nach Einflussgrößen mit Hilfe eines weiteren modellfreien Strukturidentifikationsverfahrens vor. LOLIMOT-Weiterentwicklungen mit achsschrägen Partitionierungsstrategien [4, 8] sollen ebenso untersucht werden.

Literatur

- [1] Kühn, H. „Erhaltung und Pflege von Kunstwerken“. Klinkhardt & Biermann. 2001.
- [2] Takagi, T. und Sugeno, M. „Fuzzy identification of Systems and Its Applications to Modeling and Control“. IEEE Transactions on Systems, Man and Cybernetic, (1) 116–132. 1985.
- [3] Kroll, A. und Schulte, H.: „Benchmark problems for nonlinear system identification and control using Soft Computing methods: Need and overview“. Applied Soft Computing, 496–513. 2014.
- [4] Nelles, O. „Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models“. Springer. 2001.
- [5] He, X. und Asada, H. „A New Method for Identifying Orders of Input-Output Models for Nonlinear Dynamic Systems“. American Control Conference, 2520–2523. 1993.
- [6] Kroll, A. „Computational Intelligence. Eine Einführung in Probleme, Methoden und technische Anwendungen.“. Oldenbourg Wissenschaftsverlag. 2013.
- [7] Storn, R. und Price, K. „Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces“. Journal of Global Optimization, 341–359. 1997.
- [8] Hartmann, B., Ebert, T., Fischer, T., Bel, J., Kampmann, G. und Nelles, O. „LMNtool - Toolbox zum automatischen Trainieren lokaler Modellnetze“. In: *22. Workshop Computational Intelligence*, 341–355. Hrsg.: Hoffmann, F. und Hüllermeier, E. KIT Scientific Publishing. 2012.

- 1 **BECK, S.**
Ein Konzept zur automatischen Lösung von Entscheidungsproblemen bei Unsicherheit mittels der Theorie der unscharfen Mengen und der Evidenztheorie, 2005
- 2 **MARTIN, J.**
Ein Beitrag zur Integration von Sensoren in eine anthropomorphe künstliche Hand mit flexiblen Fluidaktoren, 2004
- 3 **TRAICHEL, A.**
Neue Verfahren zur Modellierung nichtlinearer thermodynamischer Prozesse in einem Druckbehälter mit siedendem Wasser-Dampf Gemisch bei negativen Drucktransienten, 2005
- 4 **LOOSE, T.**
Konzept für eine modellgestützte Diagnostik mittels Data Mining am Beispiel der Bewegungsanalyse, 2004
- 5 **MATTHES, J.**
Eine neue Methode zur Quellenlokalisierung auf der Basis räumlich verteilter, punktwiser Konzentrationsmessungen, 2004
- 6 **MIKUT, R.; Reischl, M. (HRSG.)**
Proceedings – 14. Workshop Fuzzy-Systeme und Computational Intelligence
Dortmund, 10. - 12. November 2004
- 7 **ZIPSER, S.**
Beitrag zur modellbasierten Regelung von Verbrennungsprozessen, 2004
- 8 **STADLER, A.**
Ein Beitrag zur Ableitung regelbasierter Modelle aus Zeitreihen, 2005
- 9 **MIKUT, R.; REISCHL, M. (HRSG.)**
Proceedings – 15. Workshop Computational Intelligence
Dortmund, 16. - 18. November 2005
- 10 **BÄR, M.**
µFEMOS – Mikro-Fertigungstechniken für hybride mikrooptische Sensoren, 2005
- 11 **SCHAUDEL, F.**
Entropie- und Störungssensitivität als neues Kriterium zum Vergleich verschiedener Entscheidungskalküle, 2006
- 12 **SCHABLOWSKI-TRAUTMANN, M.**
Konzept zur Analyse der Lokomotion auf dem Laufband bei inkompletter Querschnittlähmung mit Verfahren der nichtlinearen Dynamik, 2006
- 13 **REISCHL, M.**
Ein Verfahren zum automatischen Entwurf von Mensch-Maschine-Schnittstellen am Beispiel myoelektrischer Handprothesen, 2006

- 14 **KOKER, T.**
Konzeption und Realisierung einer neuen Prozesskette zur Integration von Kohlenstoff-Nanoröhren über Handhabung in technische Anwendungen, 2007
- 15 **MIKUT, R.; REISCHL, M. (HRSG.)**
Proceedings – 16. Workshop Computational Intelligence
Dortmund, 29. November - 1. Dezember 2006
- 16 **LI, S.**
Entwicklung eines Verfahrens zur Automatisierung der CAD/CAM-Kette in der Einzelfertigung am Beispiel von Mauerwerksteinen, 2007
- 17 **BERGEMANN, M.**
Neues mechatronisches System für die Wiederherstellung der Akkommodationsfähigkeit des menschlichen Auges, 2007
- 18 **HEINTZ, R.**
Neues Verfahren zur invarianten Objekterkennung und -lokalisierung auf der Basis lokaler Merkmale, 2007
- 19 **RUCHTER, M.**
A New Concept for Mobile Environmental Education, 2007
- 20 **MIKUT, R.; Reischl, M. (HRSG.)**
Proceedings – 17. Workshop Computational Intelligence
Dortmund, 5. - 7. Dezember 2007
- 21 **LEHMANN, A.**
Neues Konzept zur Planung, Ausführung und Überwachung von Roboteraufgaben mit hierarchischen Petri-Netzen, 2008
- 22 **MIKUT, R.**
Data Mining in der Medizin und Medizintechnik, 2008
- 23 **KLINK, S.**
Neues System zur Erfassung des Akkommodationsbedarfs im menschlichen Auge, 2008
- 24 **MIKUT, R.; REISCHL, M. (HRSG.)**
Proceedings – 18. Workshop Computational Intelligence
Dortmund, 3. - 5. Dezember 2008
- 25 **WANG, L.**
Virtual environments for grid computing, 2009
- 26 **BURMEISTER, O.**
Entwicklung von Klassifikatoren zur Analyse und Interpretation zeitvarianter Signale und deren Anwendung auf Biosignale, 2009
- 27 **DICKERHOF, M.**
Ein neues Konzept für das bedarfsgerechte Informations- und Wissensmanagement in Unternehmenskooperationen der Multimaterial-Mikrosystemtechnik, 2009

- 28 **MACK, G.**
Eine neue Methodik zur modellbasierten Bestimmung dynamischer Betriebslasten im mechatronischen Fahrwerkentwicklungsprozess, 2009
- 29 **HOFFMANN, F.; HÜLLERMEIER, E. (HRSG.)**
Proceedings – 19. Workshop Computational Intelligence Dortmund, 2. - 4. Dezember 2009
- 30 **GRAUER, M.**
Neue Methodik zur Planung globaler Produktionsverbünde unter Berücksichtigung der Einflussgrößen Produktdesign, Prozessgestaltung und Standortentscheidung, 2009
- 31 **SCHINDLER, A.**
Neue Konzeption und erstmalige Realisierung eines aktiven Fahrwerks mit Preview-Strategie, 2009
- 32 **BLUME, C.; JAKOB, W.**
GLEAN. General Learning Evolutionary Algorithm and Method
Ein Evolutionärer Algorithmus und seine Anwendungen, 2009
- 33 **HOFFMANN, F.; HÜLLERMEIER, E. (HRSG.)**
Proceedings – 20. Workshop Computational Intelligence Dortmund, 1. - 3. Dezember 2010
- 34 **WERLING, M.**
Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien, 2011
- 35 **KÖVARI, L.**
Konzeption und Realisierung eines neuen Systems zur produktbegleitenden virtuellen Inbetriebnahme komplexer Förderanlagen, 2011
- 36 **GSPANN, T. S.**
Ein neues Konzept für die Anwendung von einwandigen Kohlenstoff-nanoröhren für die pH-Sensorik, 2011
- 37 **LUTZ, R.**
Neues Konzept zur 2D- und 3D-Visualisierung kontinuierlicher, multidimensionaler, meteorologischer Satellitendaten, 2011
- 38 **BOLL, M.-T.**
Ein neues Konzept zur automatisierten Bewertung von Fertigkeiten in der minimal invasiven Chirurgie für Virtual Reality Simulatoren in Grid-Umgebungen, 2011
- 39 **GRUBE, M.**
Ein neues Konzept zur Diagnose elektrochemischer Sensoren am Beispiel von pH-Glaselektroden, 2011
- 40 **HOFFMANN, F.; Hüllermeier, E. (HRSG.)**
Proceedings – 21. Workshop Computational Intelligence Dortmund, 1. - 2. Dezember 2011

- 41 **KAUFMANN, M.**
Ein Beitrag zur Informationsverarbeitung in mechatronischen Systemen, 2012
- 42 **NAGEL, J.**
Neues Konzept für die bedarfsgerechte Energieversorgung des Künstlichen Akkommodationsystems, 2012
- 43 **RHEINSCHMITT, L.**
Erstmaliger Gesamtentwurf und Realisierung der Systemintegration für das Künstliche Akkommodationssystem, 2012
- 44 **BRÜCKNER, B. W.**
Neue Methodik zur Modellierung und zum Entwurf keramischer Aktorelemente, 2012
- 45 **HOFFMANN, F.; Hüllermeier, E. (HRSG.)**
Proceedings – 22. Workshop Computational Intelligence Dortmund, 6. - 7. Dezember 2012
- 46 **HOFFMANN, F.; Hüllermeier, E. (HRSG.)**
Proceedings – 23. Workshop Computational Intelligence Dortmund, 5. - 6. Dezember 2013
- 47 **SCHILL, O.**
Konzept zur automatisierten Anpassung der neuronalen Schnittstellen bei nichtinvasiven Neuroprothesen, 2014
- 48 **BAUER, C.**
Neues Konzept zur Bewegungsanalyse und -synthese für Humanoide Roboter basierend auf Vorbildern aus der Biologie, 2014
- 49 **WAIBEL, P.**
Konzeption von Verfahren zur kamerabasierten Analyse und Optimierung von Drehrohrenprozessen, 2014
- 50 **HOFFMANN, F.; HÜLLERMEIER, E. (HRSG.)**
Proceedings. 24. Workshop Computational Intelligence, Dortmund, 27. - 28. November 2014
- 51 **EICHIN, D.**
Modellbasiertes Konzept zur vollautomatisierten Montageendprüfung von asynchron angetriebenen Getriebemotoren im lastlosen Zustand, 2015
- 52 **GRÖLL, L.**
Methodik zur Integration von Vorwissen in die Modellbildung, 2015
- 53 **IRLE, J.**
Analyse und Entwurf von Beobachtern mit unbekanntem Signalen und Parametern, 2015

54 HOFFMANN, F.; HÜLLERMEIER, E. (HRSG.)
Proceedings. 25. Workshop Computational Intelligence,
Dortmund, 26. - 27. November 2015, 2015

Die Schriften sind als PDF frei verfügbar, eine Nachbestellung der Printversion ist möglich.
Nähere Informationen unter www.ksp.kit.edu.



Dieser Tagungsband enthält die Beiträge des 25. Workshops „Computational Intelligence“ des Fachausschusses 5.14 der VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA) und der Fachgruppe „Fuzzy-Systeme und Soft-Computing“ der Gesellschaft für Informatik (GI), der vom 26. – 27. November 2014 in Dortmund stattfindet.

Der GMA-Fachausschuss 5.14 „Computational Intelligence“ entstand 2005 aus den bisherigen Fachausschüssen „Neuronale Netze und Evolutionäre Algorithmen“ (FA 5.21) sowie „Fuzzy Control“ (FA 5.22). Der Workshop steht in der Tradition der bisherigen Fuzzy-Workshops, hat aber seinen Fokus in den letzten Jahren schrittweise erweitert.

Die Schwerpunkte sind Methoden, Anwendungen und Tools für

- Fuzzy-Systeme,
- Künstliche Neuronale Netze,
- Evolutionäre Algorithmen und
- Data-Mining-Verfahren

sowie der Methodenvergleich anhand von industriellen und Benchmark-Problemen.

Die Ergebnisse werden von Teilnehmern aus Hochschulen, Forschungseinrichtungen und der Industrie in einer offenen Atmosphäre intensiv diskutiert. Dabei ist es gute Tradition, auch neue Ansätze und Ideen bereits in einem frühen Entwicklungsstadium vorzustellen, in dem sie noch nicht vollständig ausgereift sind.

