Fuzzy system identification by generating and evolutionary optimizing fuzzy rule bases consisting of relevant fuzzy rules

P. Krause, A. Krone, T. Slawinski

Faculty of Electrical Engineering, University of Dortmund, D-44221 Dortmund
Tel.: +49 231 755 4524, Fax: +49 231 755 2752
e–mail: {krause,krone,slawinski}@esr.e-technik.uni-dortmund.de

Abstract

One approach for system identification among many others is the fuzzy identification approach. The advantage of this approach compared to other analytical approaches is, that it is not necessary to make an assumption for the model to be used for the identification. In addition, the fuzzy approach can handle nonlinearities easier than analytical approaches. The Fuzzy–ROSA method is a method for data–based generation of fuzzy rules. This is the first step of a two step identification process. The second step is the optimization of the remaining free parameters, i.e. the composition of the rule base and the linguistic terms, to further improve the quality of the model and obtain small interpretable rule bases. In this paper, a new evolutionary strategy for the optimization of the linguistic terms of the output variable is presented. The effectiveness of the two step fuzzy identification is demonstrated on the benchmark problem 'kin dataset' of the Delve dataset repository and the results are compared to analytical and neural network approaches.

Keywords: fuzzy identification, fuzzy system optimization, data–based fuzzy rule generation, evolutionary strategy, Delve benchmark

1 Introduction

When trying to identify a real–world process or system there are different approaches available. These approaches can be divided into analytical approaches and CI approaches, like fuzzy logic or neural networks. The disadvantage of an analytical approach is, that an assumption for the type of model to be used for the identification has to be made. This assumption restricts the success of an applied method, because the quality of the obtained model depends strongly on the assumption. Especially, it is difficult to make assumptions on possible nonlinearities. These disadvantages are overcome by the CI approaches. Both the fuzzy logic and the neural networks need no model assumption and can handle nonlinearities more easily. Here, we use fuzzy logic, because it has the advantage of giving insight into the modeled system. The main steps of a fuzzy identification are:

- Definition of the linguistic input and output variables either by using classification methods or by heuristics.
- Generation of the rule base describing the relations between the input variables and the output variable.
- Validation and optimization of the fuzzy system.

Here, we use the Fuzzy–ROSA method [10–12] for the generation and optimization of the fuzzy system. In the first step the linguistic variables are defined and then the IF/THEN statements are set up and tested regarding to their local relevance [8, 13]. The relevant rules are incrementally collected. In the second step an optimization method is applied to the fuzzy system.

The optimization process includes two processes: First the composition of the rules is optimized. Here, a genetic algorithm searches for a good compromise between an improved quality of the model and a small number of relevant rules. In a second step the linguistic terms of the output variable are optimized. For this purpose a new approach based on an evolutionary strategy is introduced in this paper. The reasons for optimizing only the linguistic terms of the output variable are as follows: The linguistic terms of the input variables are used for the generation of the fuzzy rule base. Changing these terms withdraws the basis of the local relevance of the rules. Whereas, the definition of the linguistic terms of the output variable is mainly done by heuristics. This leads to the assumption that there is space for optimization.

In Section 2 the concept of the Fuzzy–ROSA method is briefly described. The evolutionary strategy for the optimization of the output terms is presented in Section 3. A benchmark example is discussed in Section 4. The paper closes with a short conclusion.

2 The Fuzzy–ROSA method

Main parts of the Fuzzy–ROSA method are the fuzzy rule generation and the optimization. As these parts make the core for the fuzzy identification, they are described in the following.

2.1 Concept of rule generation

The basic idea of the rule generation process is to apply a relevance test to single IF/THEN statements to assess their ability to describe a relevant aspect of the system under consideration [8, 13]. This allows not only to get transparent and comprehensible rule bases, but reduces the immense problem of finding a good rule base to the much smaller problem of finding single relevant rules.

Instead of complete rules that consider every input variable in each premise, generalizing rules are used that consider only a part of the input variables in the premise. The advantage of generalizing rules is that they cover not only one but several input situations and therefore, less rules are necessary.

For a high number of variables, an adaptive evolutionary search concept [11, 17] is used to find the relevant fuzzy rules in the search space of IF/THEN statements. An online rule reduction removes redundant rules during the rule generation process.

In comparison to methods that directly search for an optimal rule base [7, 9, 14, 15], the computing time of this approach is also practicable for applications with more than a handful variables. Even an industrial problem with 149 input variables has been successfully solved [18]. Approaches that also aim at an incremental collection of single rules [2–4] differ mainly in two points. First, they have no relevance test integrated and second, their rule search is not realized with one population in one evolutionary process.

2.2 Concepts for optimization

After a fuzzy system has been obtained there are remaining parameters, that can be used for optimization. The Fuzzy–ROSA method uses two methods:

Optimizing conflict reduction: Here, the number of rules in the fuzzy system is reduced in such a way, that the modeling error becomes as small as possible. A reduction is possible as generalizing rules are used that consider not all input variables in the premise and consequently cover each other substantially. The reduction has the further advantage of reducing the evaluation time of the fuzzy system and of making the fuzzy rule base more transparent.

Position and form of the linguistic terms of the output variable: As the definition of the linguistic terms of the output variable is mainly done by using some sort of heuristic, it can be expected that there is some space for optimization. Thus the terms are varied to decrease the modeling error under the restriction of preserving the linguistic meaning of the fuzzy rules.

3 Optimizing with an evolutionary strategy

The aim of the evolutionary strategy is to minimize the modeling error by positioning the linguistic terms of the output variable. An important restriction is, that the linguistic meaning of the fuzzy rules must be preserved. This means that the order of the terms must not change, e.g. if there are two terms $t_1 = \mathit{cold}$ and $t_2 = \mathit{hot}$ the term t_1 has to contain lower values than the term t_2. A linguistic term in the Fuzzy–ROSA method is defined as a trapezoid with four supporting points or alternatively as a singleton. In order to further support the interpretability of the rules, additional restrictions for the intervals of the terms can be given. For example if temperature is considered as an output variable, the lower bound of the terms should be -273^oC.

In contrast to other approaches in literature, that use genetic algorithms, the whole concept here is based on a standard evolutionary strategy [1,16] because the positioning of the membership functions is a real valued problem. Therefore the mutation operator, the recombination operator and the selection operator used here are standard operators. In the following two alternative representations are presented and compared, followed by a description of the fitness function.

3.1 Representation

One individual of the evolutionary strategy consists of the n_{MF} membership functions of the output variable. The $n_p(j)$ parameter[1] of the j–th membership function of the i–th individual $\vec{a}_i$ of a population are summarized in a vector $\vec{x}_{i,j}$. All vectors $\vec{x}_{i,j}$ with $j \in \{1,..,n_{MF}\}$ are consecutively represented by the individuals:

$$\vec{a}_i = \{\vec{x}_{i,1}, \vec{x}_{i,2}, .., \vec{x}_{i,n_{MF}}, \sigma_{i,1}, \sigma_{i,2}, .., \sigma_{i,n_\sigma}\} \tag{1}$$

The vectors $\vec{x}_{i,j}$ with $j \in \{1,..,n_{MF}\}$ contain the optimization parameters and the parameters $\sigma_{i,s}$ with $s \in \{1,..,n_\sigma\}$ represent the step width. Here, three possibilities exist:

- $n_\sigma = 1$ one global step width for all parameters
- $n_\sigma = n_{MF}$ one step width per membership function
- $n_\sigma = 4 \cdot n_{MF}$ one step width for each parameter[2]

There are two ways to represent the membership functions: using the positions of the supporting points or using the distances of the supporting points.

3.1.1 Position coding

Each vector[3] $\vec{x}_j$ consists of four elements describing a trapezoid. The elements have to fulfill the condition

$$\forall j \in \{1,..,n_{MF}\} : x_{j,1} \leq x_{j,2} \leq x_{j,3} \leq x_{j,4} \tag{2}$$

To get a valid individual in the sense that the order of the terms is preserved compared to the starting point the condition

$$x_{j,k} \leq x_{j+1,k} \qquad \forall k \in \{1,..,4\} \tag{3}$$

must hold. The advantage of this type of coding is that every element is independent of each other. This means a change of an element changes only one parameter of the membership function. A disadvantage is that the validity of an individual is not automatically preserved, as it is not automatically descended in the mutation and recombination process to the offspring.

3.1.2 Distance coding

If $(x_{j,1}, x_{j,2}, x_{j,3}, x_{j,4})$ are the four values describing the trapezoidal membership functions, then the k–th vector of an individual is represented by

$$\begin{aligned} \vec{x}_1 &= (x_{1,1}, d_{1,1}, d_{1,2}, d_{1,3}) \\ \vec{x}_k &= (d_{j-1}, d_{j,1}, d_{j,2}, d_{j,3}) \quad \text{if} \quad 1 < j < n_{MF}. \end{aligned} \tag{4}$$

Here $d_{j,l} = x_{j,l+1} - x_{j,l}$ is the distance between the l–th and the $(l+1)$–th supporting point of the j–th membership function and $d_j = x_{j+1,1} - x_{j,1}$ the distance between the j–th and $(j+1)$–th membership function.

[1] if using singletons $n_p(j) = 1$

[2] this assumes trapezoidal membership functions with four supporting points

[3] as only one individual is investigated the i of $\vec{x}_{i,j}$ is left out in the following

The exception is the element $x_{1,1}$ of the first membership function representing the first supporting point. To generate a valid individual, the conditions

$$\begin{array}{l} \forall j \in \{1,..,n_{MF}-1\} \ : \ d_j \geq 0 \\ \forall j \in \{1,..,n_{MF}\} \ , \ l \in \{1,2,3\} \ : \ d_{j,l} \geq 0 \end{array} \tag{5}$$

have to be fulfilled. The advantage of this coding is that after the recombination of two valid individuals a new valid individual is created. Thus, the validity of the parents is preserved. A disadvantage is that if an element of $\vec{x}_j$ is changed it has consequences for all membership functions $\vec{x}_k$ with $k > j$. This leads to undesired effects concerning the mutation operator. Therefore a repair mechanism is introduced which removes the effects on the other membership functions.

3.1.3 Comparing the representations

In addition to the form of representation, it has to be considered whether to restrict the step width or not. With the three representations, position coding, distance coding and distance coding with repair there are six cases which have to be analyzed. Figure 1 shows the results for an typical optimization process. The bottom figure is the detailed lower part of the top figure.

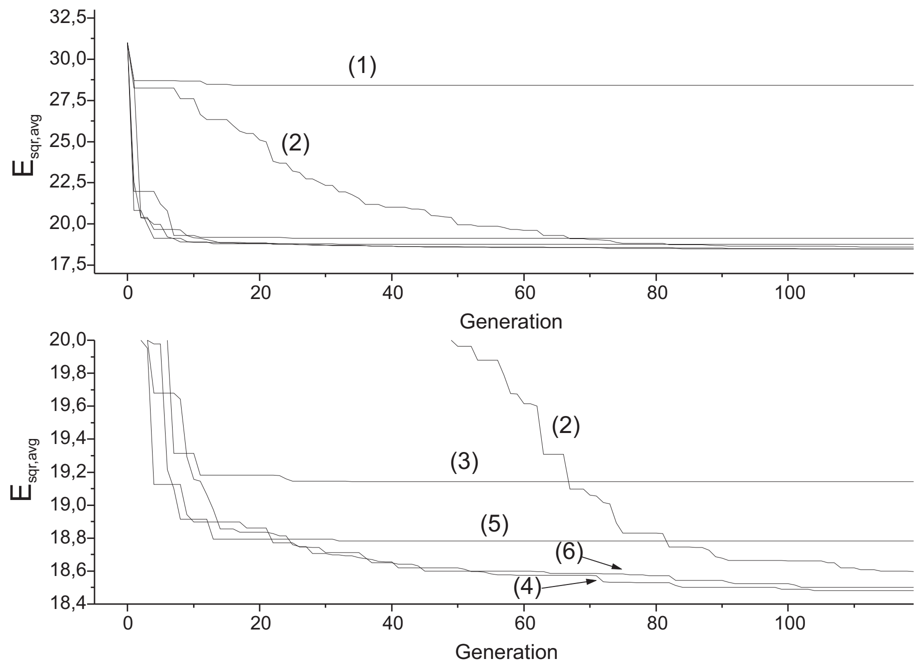

Figure 1: Best fitness over the generations using different coding variants: (1) distance coding without repair without step width restriction, (2) distance coding without repair with step width restriction, (3) position coding without step width restriction, (4) position coding with step width restriction, (5) distance coding with repair and without step width restriction, (6) distance coding with repair and with step width restriction.

The results show that the optimization with step width restriction converges at a better fitness than the optimization without step width restriction. The optimizations with position coding and distance coding with repair show the best behavior for the convergence time and for the reached fitness.

3.2 Fitness

The fitness evaluation is done in two steps. First the modeling error is evaluated and second a penalty factor is applied. For the evaluation of the modeling error the output values of the fuzzy system are compared to the original output values of the training data. This can be done either by using the average absolute error or the average square error. If the individual is invalid in the sense that one of the coding conditions is violated or that an interval restriction of a membership functions is violated, a penalty factor proportional to the degree of violation is applied to the individual.

4 Example

As an example, the 'kin dataset' of the Delve[4] dataset repository is chosen. The kin datasets are a family of datasets synthetically generated from a realistic simulation of the forward kinematics of an 8 link all–revolute robot arm. The tasks associated with these datasets consist of predicting the distance of the end–effector from a target, given the 8 angular positions of the joints and also given the link twist angles, link lengths, and link offset distances for all 8 joints resulting in 32 input variables altogether. The dependency between the input variables and the output variables is nonlinear and noisy.

The available tasks of the kin dataset differ in the degree of linearity, the amount of noise and the number of input variables. In the following the task kin-32nm (32 input variables, highly nonlinear, moderate noise) is considered. For each input variable and the output variable seven equi–distant distributed linguistic terms are defined. The range of the output variable spans from zero to three. The modeling error is measured as the average absolute error. In a first step only rules are generated with a combination depth[5] of one. After the generation the fuzzy system was optimized with the methods presented before. The results are shown in table 1.

Table 1: Results for kin-32nm with the Fuzzy–ROSA method with a combination depth of one

	number of rules	error rate on training data	error rate on validation data
	185	0.344	0.343
optimized	74	0.311	0.329

In a second step the combination depth was increased by one and the same steps as before were carried out. The results are shown in table 2. It can be seen that the initial results after the generation are better than the results achieved before. After the optimization the results for the training data is further improved, but the performance on the validation data decreases. This is the common problem of overfitting, which could be prevented by a method for early stopping.

Table 2: Results for kin-32nm with the Fuzzy–ROSA method with a combination depth of two

	number of rules	error rate on training data	error rate on validation data
	2892	0.303	0.322
optimized	189	0.278	0.338

These results are compared to the results of other methods used for solving the kin problem. The following methods have been chosen:

gp-map-1 A Gaussian processes for regression trained with a maximum–aposteriori approach implemented with conjugate gradient optimization [19].

mars3.6-bag-1 Multivariate Adaptive Regression Splines (MARS) method developed by Jerome Friedman. The bagging procedure is used in conjunction with MARS [5].

knn-cv-1 This method uses an average of the k nearest neighbors in the training set for predictions. For each loss type, the value of k (the neighborhood size) is chosen leave–one–out cross validation, repeated exhaustively for all possible values of k.

[4]Data for Evaluating Learning in Valid Experiments *http://www.cs.utoronto.ca/~delve/index.html*

[5]combination depth is the number of linguistic expressions in the premise

mlp-mdl-vh Minimum description length (mdl) based training of a multilayer perceptron (mlp) (feedforward neural network) with a single layer of hidden units [6].

The table 3 shows the average absolute error on the validation data of the different methods.

Table 3: Comparison of different methods

mlp-mdl-vh	gp-map-1	mars3.6-bag-1	Fuzzy–ROSA	knn-cv-1
0.2979	0.2996	0.3281	0.329	0.3327

5 Conclusion

In this paper we presented how a fuzzy approach can be applied for system identification. Along with the description of the Fuzzy–ROSA method we presented a new evolutionary strategy for the optimization of the position of the linguistic terms of the output variable. Here, it has been shown that the selection of the representation for the individuals is the key to a good performance of the optimization. The methods were applied to the benchmark problem 'kin dataset'. The results are in the scale of the results of other analytical and neural network methods. The advantage of the fuzzy approach is that the result consists of only 74 interpretable fuzzy rules. This is remarkable because the problem consists of 32 input variables with each having seven linguistic terms. Further work is necessary to design an intelligent method for early stopping for preventing an overfitting on the training data.

Acknowledgement:
This research was sponsored by the Deutsche Forschungsgemeinschaft (DFG), as part of the Collaborative Research Center 'Computational Intelligence' (531) of the University of Dortmund.

Part of the **references** can be **downloaded** from
http://esr.e-technik.uni-dortmund.de/winrosa/winrosa.htm.

References

[1] T. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, NY, 1996.

[2] A. Bonarini. Evolutionary learning of general fuzzy rules with biased evaluation functions: Competition and cooperation. In *Proceedings of the 1994 IEEE World Congress on Computational Intelligence, WCCI '94*, pages 51–56, 1994.

[3] O. Cordon, A. Gonzalez, F. Herrera, and R. Perez. Encouraging cooperation in the genetic iterative rule learning approach for qualitative modeling. In *Computing with Words in Information/Intelligent Systems*, pages 95 – 117, Heidelberg, 1999. Physica.

[4] O. Cordon, F. Herrera, and M. Lozano. A three–stage method for designing genetic fuzzy systems by learning from examples. In *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, PPSN IV*, pages 720–729, Berlin, 1996. Springer.

[5] J. H. Friedmann. Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, 19:1–141, 1991.

[6] E. Hinton, G. and D. van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Sixth Annual Conference on Computational Learning Theory*, pages 5–13, 1993.

[7] C.-C. Hsu, S.-I. Yamada, H. Fujikawa, and K. Shida. A fuzzy self–tuning parallel genetic algorithm for optimization. *Computers and Industrial Engineering*, 30:883–893, 1996.

[8] H. Jessen and T. Slawinski. Test– and rating strategies for data–based rule generation. In *Computational Intelligence.* CI–39/98, Collaborative Research Center 531, University of Dortmund, 1998.

[9] Y. Jin, W. von Seelen, and B. Sendhoff. An approach to rule–based knowledge extraction. In *Proceedings of the Seventh IEEE International Conference on Fuzzy Systems, FUZZ–IEEE '98*, volume 2, pages 1188–1193, Anchorage, Alaska, USA, 1998.

[10] P. Krause. Demonstration of the use of hyperinference: Position control using an electric-disc motor. In B. Reusch, editor, *Proc 6th Int'l Conf. on Computational Intelligence (6th Fuzzy-Days)*, pages 671–675, Berlin Heidelberg, 1999. Springer-Verlag.

[11] A. Krone and H. Kiendl. Evolutionary concept for generating relevant fuzzy rules from data. *International Journal of Knowledge–based Intelligent Engineering Systems*, 1(4):207–213, 1997.

[12] A. Krone and U. Schwane. Generating fuzzy rules from contradictory data of different control strategies and control performances. In *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, FUZZ–IEEE '96*, volume 1, pages 492–497, New Orleans, USA, 1996.

[13] A. Krone and H. Taeger. Relevance test for fuzzy rules. In *Computational Intelligence*. CI–40/98, Collaborative Research Center 531, University of Dortmund, 1998.

[14] O. Nelles, M. Fischer, and B. Müller. Fuzzy rule extraction by a genetic algorithm and constrained nonlinear optimization of membership functions. In *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, FUZZ–IEEE '96*, volume 1, pages 213–219, New Orleans, USA, 1996.

[15] D. Popovic and N. Xiong. Design of flexible structured fuzzy controllers using genetic algorithms. In *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, FUZZ–IEEE '96*, volume 3, pages 1682–1686, New Orleans, USA, 1996.

[16] H.-P. Schwefel. *Evolution and optimum seeking*. Sixth–Generation Computer Technology Series, New York, USA, 1995.

[17] T. Slawinski, A. Krone, U. Hammel, D. Wiesmann, and P. Krause. A hybrid evolutionary search concept for data-based generation of relevant fuzzy rules in high dimensional spaces. In *Proceedings of Eighth International Conference on Fuzzy Systems, FUZZ-IEEE'99*, Korea, 1999.

[18] T. Slawinski, J. Praczyk, U. Schwane, A. Krone, and H. Kiendl. Data–based generation of fuzzy–rules for classification, prediction and control with the Fuzzy–ROSA method. In *European Control Congress, ECC '99*, Karlsruhe, 1999.

[19] K. I. Williams, C. and E. Rasmussen, C. Gaussian processes for regression. In M. E. Hasselmo D. S. Touretzky, M. C. Mozer, editor, *Advances in Neural Information Processing Systems*. MIT Press, 1996.