New Initial Fuzzy System Generation Features in the SFMI Toolbox

Zsolt Csaba Johanyák
Kecskemét College/Department of Information Technologies, Kecskemét, Hungary
johanyak.csaba@gamf.kefo.hu

***Abstract*—The sparse fuzzy model identification (SFMI) toolbox is a Matlab based software that was developed to facilitate the creation of fuzzy systems with a compact and low complexity rule base. The objective of this paper to report some extensions and new features of the toolbox that aim the widening of the pool of the applicable approaches and methods in the process of automatic rule base creation from sample data.**

I. Introduction

The functioning of a fuzzy inference based system is determined by its rule base and the applied inference method. The rule base can be compiled manually based entirely on human knowledge and expertise and/or using methods (software) that allow the automatic rule extraction and parameter tuning using machine learning approaches. The first version of the SFMI toolbox [11] divided the model identification into two steps: (1) creation of an initial rule base, (2) tuning of the parameters. The later one could also include the creation of new rules (extension of the rule base) [15] as well.

In this paper, we review briefly the initial rule base creation related features that were present in the original version of the toolbox and we give an in-details report of the two new methods for the creation of the initial rule base.

The rest of this paper is organized as follows. Section II reviews the applicable initial rule base generation methods. Section III presents the applicable graphical representation modes. Section IV gives a short review of the applicable fuzzy inference techniques. Section V presents the internal representation of the generated fuzzy systems and the conclusions are drawn in Section VI.

II. Creation of the Initial Rule Base

Initially [11] the SFMI toolbox supported only two kinds of starting rule base creation: one based on fuzzy clustering and one generating two rules that correspond to the lowest and highest output of the sample data. Recently this was extended by the so called “2^n+2” approach and a grid partition based solution. Further on we will recall briefly the key ideas of the two original methods followed by the presentation of the two new solutions. We use the synthetic function (see Figure 1)

$$z = x^3 - 3\cdot y^3 + 100\cdot y^2 + 40$$
$$x \in [0, 50],\ y \in [0,50] \quad (1)$$

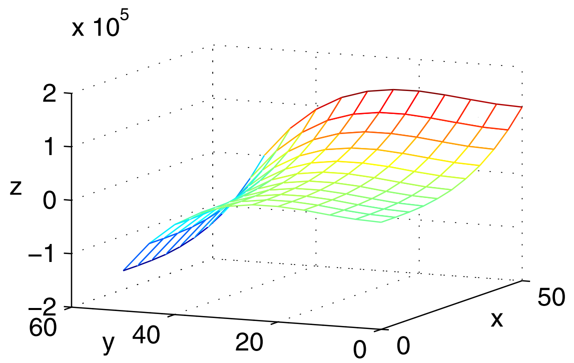

Figure 1. Surface described by the synthetic function used for the sample data generation

for illustration purposes, i.e. the sample data for which the software will create different fuzzy systems is generated using this function.

A. Fuzzy Clustering Based Initial Rule Base

The fuzzy clustering based rule extraction is a common approach in sample data based fuzzy model identification [2][3][9][25][26][29]. The implementation in the SFMI toolbox follows the method called automatic fuzzy system generation based on fuzzy clustering and projection (ACP) [5]. It starts with a one dimensional clustering of the sample output data. The membership functions created from the clusters are trapezoidal and their parameters are determined by a horizontal cut of the clusters at a predefined α-level. The α-cut endpoints define the core of the fuzzy sets; while the support is determined by the

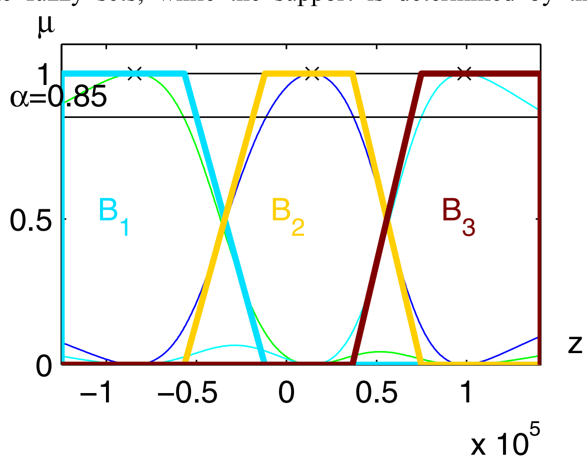

Figure 2. Output clusters and membership functions with a cut level α=0.85

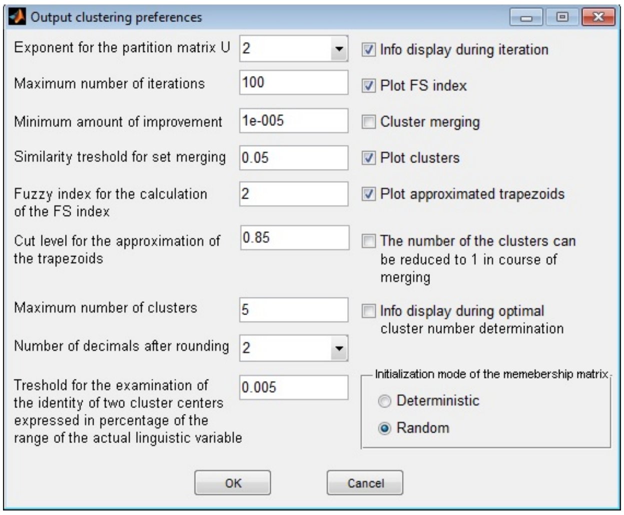

Figure 3. Parameterization of the output clustering and fuzzy membership function extraction

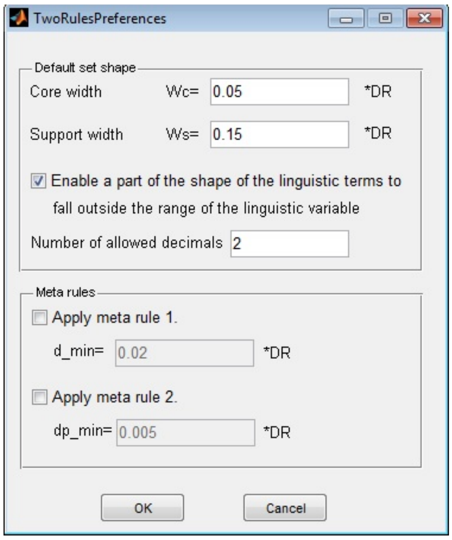

Figure 5. Parameterization of the method that creates two initial rules

Ruspini character of the partition (see Figure 2).

Next, for each fuzzy set one selects the corresponding data rows from the sample data which is followed by a one dimensional fuzzy c-mean clustering of the selected data in each antecedent dimension. This step is also called projection into the antecedent space.

In each antecedent dimension the obtained cluster centers are collected into a single group and are examined against some cluster mergeability conditions. The final fuzzy sets are generated from the remaining clusters using the same horizontal cut based technique as in the case of the consequent partition.

Having the antecedent and consequent partitions generated the rules are created based on the sample data rows that correspond to the individual output fuzzy sets. Figure 4 depicts the antecedents of the rules in case of the above presented sample data set.

Parameters of the clustering and membership function extraction are (see e.g. Figure 3): exponent of the partition matrix U, maximum allowed number of iterations, minimum amount of objective function improvement in case of two consecutive iteration cycles, fuzzy index for the calculation of the optimal cluster number, maximum number of clusters, permission of the cluster merging, threshold for the examination of the identity (mergeability) of two clusters, and the similarity threshold for set merging in the antecedent dimensions.

B. System with two initial rules

The "two initial rules" based method [15] first selects two data rows from the sample data set: one that represents the lowest output value, and one that contains the highest output value. If more than one data row contains the same extreme output value that one is selected which is closer to the lower or upper bound of the antecedent space. Next, two rules are created that describe the two specified output values.

The fuzzy sets contained in the antecedent and consequent parts of the rules are created with trapezoidal membership functions where the width of the core (w_c) and support (w_s) are defined in function of the range of the actual partition and the reference point is identical with the respective element of the selected data row. Optionally one can select the application of the following two meta rules as well.

1. If the distance between the reference point of the new fuzzy set and the reference point of an old fuzzy set is smaller or equal than a specified threshold value

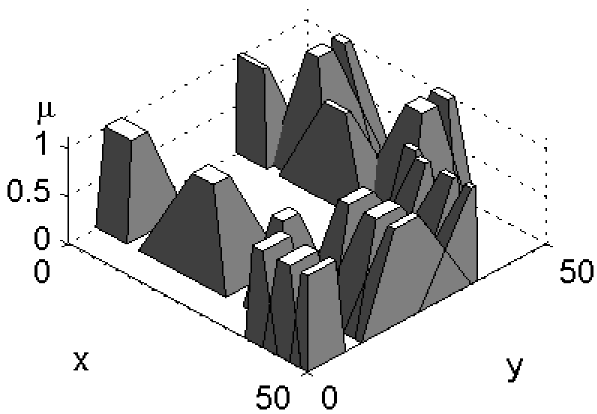

Figure 4. Rule antecedents of the rule base obtained by fuzzy clustering based approach

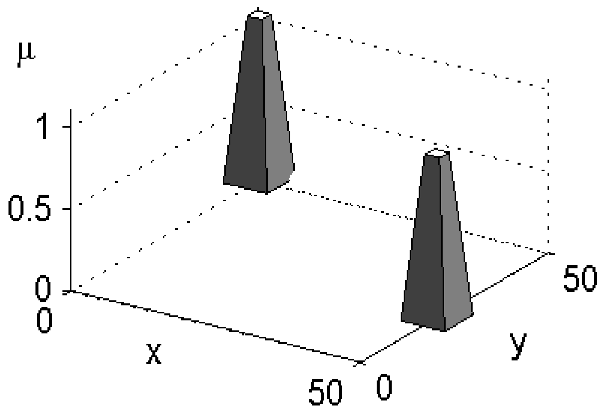

Figure 6. Rule antecedents of the rule base obtained by "two rules" method

(d_{min}), the old set is used as antecedent/consequent of the new rule in the current input/output dimension (the algorithm does not create a new fuzzy set).

2. If the average distance between the corresponding parameters of the new set and an old set is smaller or equal than a specified threshold (dp_{min}) the two sets are merged into one.

Figure 6 depicts the antecedents of the two rules in case of the above presented sample data set. The method has six main parameters: two for the definition of the membership functions (the coefficients for w_c and w_s), two for the meta rules (the coefficients for d_{min} and dp_{min}), and two more parameters that specify whether the meta rules should be applied.

C. System with 2^n+2 rules

The rule base generation method that applies the above presented "two rules" approach usually ensures after the parameter optimization a final fuzzy system with a low number of rules; however, the process can be very time consuming. Besides, another problem could arise from the deficiency that not all the FRI methods have extrapolation capability. In order to alleviate the above mentioned problems the SFMI toolbox offers as a new feature an initial rule base generation mode that beside the above specified two rules also creates a rule for each corner point of the hypercube of the antecedent space.

This approach results in (at most) four rules in case of a one dimensional antecedent space, and ten rules in case of a two dimensional antecedent space. In the general case the number of rules will be

$$N_R = 2^n + 2, \qquad (2)$$

where n is the number of input dimensions. Sometimes the actual number of rules can be less than the value specified in (2) because one or both of the extreme outputs could correspond to corner-points of the antecedent hypercube. For example in case of our synthetic data set one gets a rule base with only five rules because the rule describing the corner point x=0, y=50 is identical with the rule corresponding to the lowest output. The main steps of the algorithm are presented below.

1. Create an initial rule base with the "two initial rules" method.
2. Create a 2 by n matrix with the lower (row 1) and upper (row 2) bounds of the input dimensions.
3. Create a 2^n by n+1 matrix with the coordinates of the antecedent space's corner points (first n columns) and the respective output values (n+1th) column.
4. For each corner point
   a. In each antecedent dimension
      i. IF there is no fuzzy set with the same reference point as the corner point THEN create a new fuzzy set.
   b. In the consequent dimension
      i. IF there is no fuzzy set with the same reference point as the output value THEN create a new fuzzy set.
   c. Create a rule that describes the current corner point and the corresponding output.
   d. IF the current rule is a duplicate THEN remove the current rule.
   e. IF meta rule 2 is allowed THEN apply it.

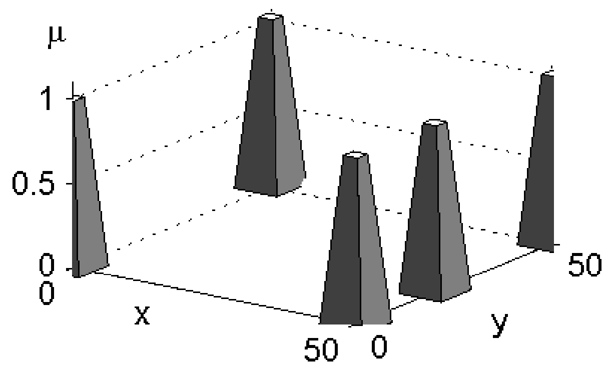

Figure 7. Rule antecedents of the rule base obtained by "2^n+2" method

The first two rules and the fuzzy sets of the other rules as well are generated using the technique presented in Section II.B. The parameters of the method are also identical with the previously presented ones. The "2^n+2" approach can be used only when all the corner points of the antecedent space are covered by sample data.

D. System Based on Grid Partitioning

All the three previously presented methods create a sparse rule base, i.e. for some possible observations (allowed input values) there are no applicable rules. As an extension of its functionality the 1.1.6 version of the SFMI toolbox also supports the creation of a rule base that ensures a full coverage of the input space allowing the system architect the specification of the coverage level, the number of the fuzzy sets in the current partition, and the width of the fuzzy sets' core as well (see Figure 8).

The algorithm defines the characteristics of the grid partitioning conform Figure 9, where μ_c is the coverage level ($\mu_c \in [0.1, 1]$),

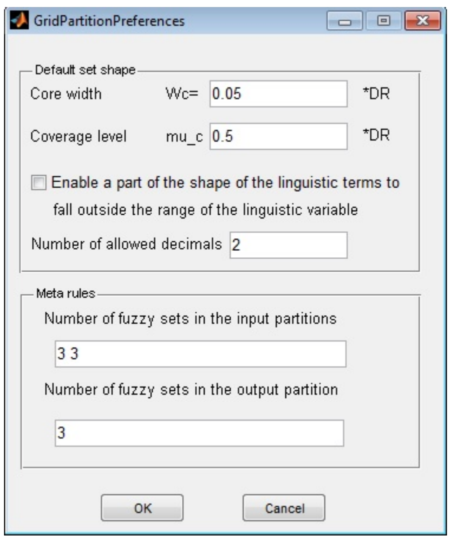

Figure 8. Parameterization of the grid partition based method

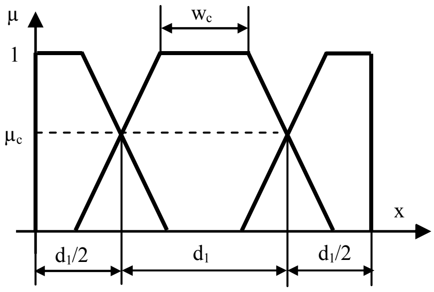

Figure 9. Characteristics of the grid partitioning in the current partition

$$d_1 = \frac{DR}{n_s - 1}, \tag{3}$$

n_s is the number of the fuzzy sets in the partition, and DR is the range of the partition. Using core center type reference points the distance between the reference points of the neighboring fuzzy sets will be also d_1.

The width of the fuzzy sets' core (w_c) is defined by the user in proportion of DR by the means of the coefficient c_c

$$w_c = c_c \cdot DR, \quad c_c \in \left[0, \frac{d_1}{DR}\right], \tag{4}$$

where the upper limit for c_c is determined by the number of fuzzy sets in the partition.

The width of the fuzzy sets' support depends on two factors that are w_c and μ_c. From Figure 10 we can write the equation

$$\frac{\frac{w_s}{2} - \frac{d_1}{2}}{\frac{w_s}{2} - \frac{w_c}{2}} = \frac{\mu_c}{1}. \tag{5}$$

Thus the width of the support will be

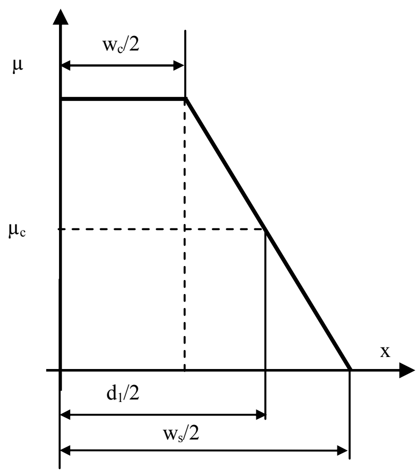

Figure 10.

$$w_s = \frac{d_1 - \mu_c \cdot w_c}{1 - \mu_c}. \tag{6}$$

The resulting partition has a Ruspini character if μ_c=0.5. Having core center type reference points the abscissa value corresponding to the ith reference point will be

$$RP_{xi} = x_{\min} + (i-1) \cdot d_1, \tag{7}$$

where x_{min} is the lower bound of the current partition. The number of rules necessary for the full coverage of the input space is

$$N_R = \prod_{i=1}^{n} n_{si}, \tag{7}$$

n_{si} is the number of the fuzzy sets in the ith antecedent dimension.

The main steps of the algorithm are presented below.

1. In each antecedent dimension
   a. Calculate d_1, w_c, and w_s.
   b. Create partition.
2. In the consequent dimension
   a. Calculate d_1, w_c, and w_s.
   b. Create partition.
3. For each possible antecedent set combination (N_R)
   a. Select the data rows from the sample that fall inside the hypercube defined by the supports of the related fuzzy sets.
   b. In case of each selected data row find the consequent set(s) where the output of the data row falls inside the support.
   c. Select the consequent set with the most data rows.
   d. Create a rule with the selected consequent set.

III. Graphical Representation

The SFMI toolbox supports three types of graphical representation. The first one is the classical two-dimensional representation of a partition's membership functions. For example Figure 11 shows the output

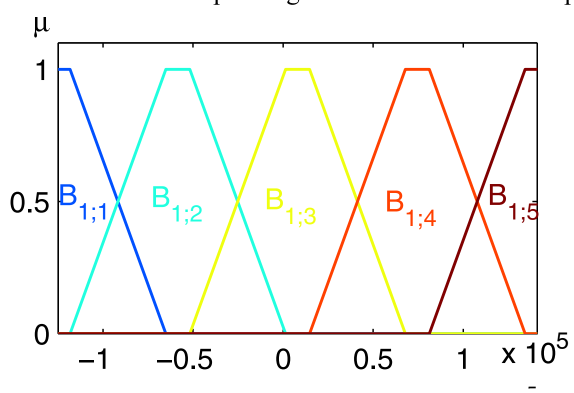

Figure 11. Output membership functions in case of grid partitioning

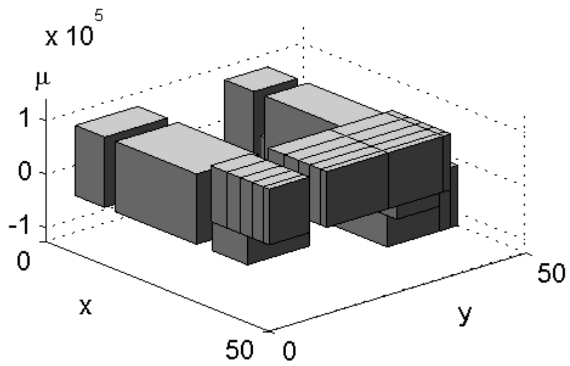

Figure 12. 3D representation of the rule base obtained by clustering based method

partition of the fuzzy system generated using the grid partitioning based approach with five fuzzy sets in a partition.

The second type of graphical representation is a three-dimensional one and is applicable either in case of 1D in – 1D out or in case of 2D in – 1D out fuzzy systems. In the former case it shows the rules in form of bodies defined by the related antecedent and consequent sets (for example truncated pyramids in case of trapezoidal membership functions on both the antecedent and consequent side). In the latter case (see e.g. Figure 4) the body represents the antecedent part of a rule.

The third type of graphical representation gives a three-dimensional picture of a 2D in – 1D out fuzzy system's rule base. Here each rule is represented by a brick whose edges are defined by the supports of the fuzzy sets involved in the rule. For example Figure 12 represents the rule base of the fuzzy system obtained by the clustering based method.

IV. Fuzzy Inference

In course of the parameter identification one also needs the evaluation of a given fuzzy system, which is based on fuzzy inference in a (usually) sparse rule base. The SFMI toolbox instead of containing a built-in inference engine it relies on the reasoning services provided by the fuzzy rule interpolation (FRI) toolbox [16], which is also a free Matlab based software available at [7]. It contains the implementation of the following FRI techniques: CRF [6], FIVE [19], FRIPOC [12], FRISUV [8], IMUL [30], KH [17], LESFRI [13], MACI [1], VEIN [14], VKK [28]. Further plans include the implementation of other FRI methods (e.g. [4][18]) as well.

V. Internal Representation of the Generated Fuzzy System

The internal representation of the generated fuzzy system is identical with the one used by the FRI toolbox. It is an extended version of the Matlab Fuzzy Logic ToolBox's (FL TB) internal FIS representation. It is easy understandable, well-arranged, and based on series of embedded structures. Figure 13 presents the most relevant elements of this data structure. The "(*i*)" index indicates that the respective field is a vector. The structure contains some fields storing general information (e.g. the name of the system (*name*), the type of the used reference point (*refType*), etc.).

```
fis--name
    |-type
    |-...
    |-refType
    |
    |-input(i)--name
    |          |-range(i)
    |          |-mf(i)--name
    |                 |-type
    |                 |-params(i)
    |                 |-paramsy(i)
    |
    |-output(i)--name
    |           |-range(i)
    |           |-mf(i)--name
    |                  |-type
    |                  |-params(i)
    |                  |-paramsy(i)
    |
    |-rule(i)—antecedent(i)
             |-consequent(i)
             |-weight
             |-connection
```

Figure 13. Data structure used by the FRI TB for fuzzy inference system representation

The *input* vector-field describes the antecedent part of the system. Each of its elements is a structure that stores a partition. This structure contains three fields, one for the name of the current linguistic variable (*name*), a vector with two elements for the lower and upper bounds of the partition (*range*), and one with the membership functions of the partition (*mf*). The last one is also a vector that contains a separate structure for each fuzzy set. This structure has four fields, which are the name of the current fuzzy set (*name*), its type (*type*), and two other fields for the parameters. In case of the most used piece-wise linear membership functions (e.g. trapezoidal, triangle shaped, etc.) the abscissa (*params*) and ordinate (*paramsy*) values of the break-points are stored as parameters.

The *output* field of the *fis* structure describes the consequent part of the FIS in an identical way with the *input* field. The rule base is stored by the help of the *rule* field. Each element of this vector describes a rule with a structure, which contains four fields. The first one (*antecedent*) is a vector having the same size as the number of antecedent dimensions. It stores in case of each input dimension the ordinal number of the fuzzy sets belonging to the premise of the rule. The second field (*consequent*) holds the ordinal numbers of the rule consequents' sets in a similar way. The last two fields (*weight* and *connection*) stand for the weights associated with the rules, respectively the type of connections between the sets of a rule (AND, OR). For now they are not used by FRI TB, they are kept for compatibility purposes.

VI. Conclusions

The SFMI toolbox provides an easy-to-use service for the generation of sparse and covering (dense) rule bases. It has a wide application area: it can be used for tuning of different kinds of fuzzy controllers [24][22], decision making systems [27], expert systems [5][23], risk assessment systems [21], etc.

In this paper we gave a review of the toolbox's new and previously existent features that are related to the generation of the initial rule base where the main enhancement was the inclusion of the two new methods for the generation of the initial rule base. This rule base will be tuned later using a parameter optimization method.

Further development plans of the toolbox include the implementation of new optimization methods (e.g. [31]) as well as the optional applicability of the vector based internal fuzzy system representation (VFIS) introduced in [10]. The use of VFIS can contribute significantly to the decrease of the fuzzy inference's time need in case of sparse rule bases.

ACKNOWLEDGMENT

This research was supported by the National Scientific Research Fund Grant OTKA K77809.

REFERENCES

[1] P. Baranyi, D. Tikk, Y. Yam, L. T. Kóczy and L. Nádai A New Method for Avoiding Abnormal Conclusion for alpha-cut Based Rule Interpolation, 8th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'99), Seoul, Korea, 22-25August, 1999, pp. 383-388.

[2] Chong, A.: Constructing Sparse and Hierarchical Fuzzy Rulebases, PhD Thesis, Murdoch University, Perth, W.A., 2004.

[3] Chong, A., Gedeon, T. D. and Kóczy, L. T.: Projection Based Method for Sparse Fuzzy System Generation, in Proceedings of the 2nd WSEAS International Conference on Scientific Computation and Soft Computing, Crete, Greece, 2002, pp. 321-325.

[4] M. Detyniecki, Ch. Marsala, M. Rifqi : "Double-linear fuzzy interpolation method", IEEE International Conference on Fuzzy Systems, FUZZIEEE'2011, Taipei, ROC, pp. 455-462 (2011)

[5] Devasenapati, S.B., Ramachandran, K.I.: Hybrid Fuzzy Model Based Expert Sytem for Misfire Detection in Automobile Engines, International Journal of Artificial Intelligence, Vol. 7, No. A11, pp. 47--62 (2011)

[6] Gedeon, T. D., Kóczy, L. T.: Conservation of fuzziness in rule interpolation. In Proceedings of the Symposium on New Trends in Control of Large Scale Systems, Volume 1, Herl'any, Slovakia, 1996, pp. 13-19.

[7] http://fri.gamf.hu

[8] Z.C. Johanyák: Fuzzy Rule Interpolation based on Subsethood Values, in Proceedings of 2010 IEEEIntereational Conference on Systems Man, and Cybernetics (SMC 2010), 10-13 October 2010, ISBN 978-1-424-6587-3, pp. 2387-2393.

[9] Johanyák, Z.C.: Fuzzy rule interpolation methods and automatic system generation based on sample data (in Hungarian), PhD thesis, University of Miskolc, 2008.

[10] Z.C. Johanyák: Performance Improvement of the Fuzzy Rule Interpolation Method LESFRI, in Proceeding of the 12th IEEE International Symposium on Computational Intelligence and Informatics, Budapest, Hungary, November 21-22, 2011, pp. 271-276.

[11] Johanyák, Z.C.: Sparse Fuzzy Model Identification Matlab Toolbox - RuleMaker Toolbox, IEEE 6th International Conference on Computational Cybernetics, November 27-29, 2008, Stara Lesná, Slovakia, pp. 69-74.

[12] Johanyák, Z.C., Kovács S.: Fuzzy Rule Interpolation Based on Polar Cuts, Computational Intelligence, Theory and Applications, Springer Berlin Heidelberg, 2006, ISBN 978-3-540-34780-4, pp. 499-511.

[13] Johanyák, Zs. Cs., Kovács, Sz.: Fuzzy Rule Interpolation by the Least Squares Method, 7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI 2006), November 24-25, 2006 Budapest, ISBN 963 7154 54 X, pp. 495-506.

[14] Johanyák, Zs. Cs., Kovács, Sz.: Vague Environment-based Two-step Fuzzy Rule Interpolation Method, 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics (SAMI 2007), January 25-26, 2007 Poprad, Slovakia, ISBN 978 963 7154 56 0, pp. 189-200.

[15] Johanyák, Z.C., Kovács, S.: Sparse Fuzzy System Generation by Rule Base Extension, 11th IEEE International Conference of Intelligent Engineering Systems (IEEE INES 2007), June 29 - July 1, 2007, Budapest, pp. 99-104.

[16] Z.C. Johanyák, D. Tikk, S. Kovács, K. W. Wong: "Fuzzy Rule Interpolation Matlab Toolbox – FRI Toolbox," Proc. of the IEEE World Congress on Computational Intelligence (WCCI'06), 15th Int. Conf. on Fuzzy Systems (FUZZ-IEEE'06), July 16--21, 2006, Vancouver, BC, Canada, pp. 1427-1433.

[17] L. T. Kóczy, K. Hirota, 1992: Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases. In Information Sciences 71, 1992. pp 169-201. ISSN 0020-0255

[18] Z. Krizsán, Sz. Kovács Double Fuzzy Dot Extension of the FRIPOC Fuzzy Rule Interpolation Method, 4th IEEE International Symposium on Logistics and Industrial Informatics (LINDI 2012), IEEE, pp. 191-196, 2012

[19] Kovács, Sz.: Extending the Fuzzy Rule Interpolation "FIVE" by Fuzzy Observation, Advances in Soft Computing, Computational Intelligence, Theory and Applications, Bernd Reusch (Ed.), S pringer Germany, ISBN 3-540-34780-1, pp. 485-497, (2006).

[20] Perfilieva, I., Wrublova, M. Hodakova, P.: Fuzzy Interpolation According to Fuzzy and Classical Conditions , Acta Polytechnica Hungarica, Vol. 7, Issue 4, Special Issue: SI, pp. 39--55 (2010)

[21] Portik T., Pokorádi L (2013), "Fuzzy rule based risk assessment with summarized defuzzyfication,", In Proceedings of the XIIIth Conference on Mathematics and its Applications. , pp. 277–282.

[22] R.E. Precup, S. Preitl, E. M. Petriu, J. K. Tar, M. L. Tomescu and C. Pozna (2009): Generic two-degree-of-freedom linear and fuzzy controllers for integral processes, Journal of The Franklin Institute, vol. 346, no. 10, pp. 980-1003, Dec. 2009.

[23] O.A. Schipor, S.G. Pentiuc, M.D. Schipor: Improving Computer Based Speech Therapy Using A Fuzzy Expert System, Computing and Informatics, Vol. 29, 2010, 303–318.

[24] I. Škrjanc, S. Blažič, S. Oblak and J. Richalet (2004): An approach to predictive control of multivariable time-delayed plant: Stability and design issues, ISA Transactions, vol. 43, no. 4, pp. 585-595, Oct. 2004.

[25] Sugeno, M. and Yasukawa, T.: A fuzzy-logic-based approach to qualitative modeling, in IEEE Transactions on Fuzzy Systems, Vol. 1, 1993, pp. 7-31.

[26] Tikk, D., Gedeon, T. D. Kóczy, L. T. and Bíró, G.: Implementation details of problems in Sugeno and Yasukawa's qualitative modeling, Research Working Paper RWP-IT-02-2001, School of Information Technology, Murdoch University, Perth, W.A., 2001.

[27] Ján Vaščák (2011): Decision-making Systems in Mobile Robotics, In: Autonomous Decision Systems Handbook, BEN, Prague, Czech Republic, pp. 56-88, ISBN 978-80-7300-415-6.

[28] G. Vass, L. Kalmár, L. T. Kóczy, "Extension of the fuzzy rule interpolation method", in Proc. Int. Conf. Fuzzy Sets Theory Applications (FSTA '92), Liptovsky M., Czechoslovakia, 1992, pp. 1-6.

[29] Wong, K. W., Kóczy, L. T., Gedeon, T. D., Chong, A. and Tikk, D. :Improvement of the Cluster Searching Algorithm in Sugeno and Yasukawa's Qualitative Modeling Approach, in Lecture Notes in Computer Science, Vol. 2206, 2001, pp. 536–549.

[30] K. W. Wong, D. Tikk, T. D. Gedeon and L. T. Kóczy, "Fuzzy rule interpolation for multidimensional input spaces with applications: A case study". IEEE Trans of Fuzzy Systems , 13(6), pp. 809–819, December 2005.

[31] K. Balázs, J. Botzheim, L.T. Kóczy: Comparison of Various Evolutionary and Memetic Algorithms, Integrated Uncertainty Management and Applications, Advances in Intelligent and Soft Computing, Springer Berlin Heidelberg, Vol. 68, 2010, pp. 431-442.