Volume 44, Number 2, 2003

ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

AUTOMATIC REMOVAL OF NOISY DATA FROM DATA SETS. APPLICATION IN CIRCUIT FUNCTIONS MODELING

Gabriel OLTEAN, Mihai CRASI
Technical University of Cluj Napoca, Faculty of Electronics and Telecomunications, Basis of Electronics Department
Daicoviciu 15, Tel:195699 ext:186 email:{ Gabriel.Oltean,Mihai.Crasi}@bel.utcluj.ro

Abstract: The problem to build a model of an unknown complex function from data sets is a fundamental issue in a variety of scientific and engineering fields. Even in the case of a large amount of data, there can be some situations for that some representative features of the function are altered due to the noisy measurement. In order to obtain an accurate model we have developed and implemented a method to automatically find and eliminate the data points affected by noise. The interaction between the user and the computer program is facilitated by a friendly graphical user interface.
We validated the method by using it to model two bi-dimensional mathematical function: a linear one and a nonlinear one. As an application of the method we use it to model a circuit function for a simple operational transconductance amplifier..

Key words: *data set, noise cancellation, fuzzy model, analog circuit function*
.

AUTOMATIC REMOVAL OF NOISY DATA FROM DATA SETS. APPLICATION IN CIRCUIT FUNCTIONS MODELING

Gabriel OLTEAN, Mihai CRASI
Technical University of Cluj Napoca, Faculty of Electronics and Telecomunications, Basis of Electronics Department
Daicoviciu 15, Tel:195699 ext:186 email:{ Gabriel.Oltean,Mihai.Crasi}@bel.utcluj.ro

Abstract: The problem to build a model of an unknown complex function from data sets is a fundamental issue in a variety of scientific and engineering fields. Even in the case of a large amount of data, there can be some situations for that some representative features of the function are altered due to the noisy measurement. In order to obtain an accurate model we have developed and implemented a method to automatically find and eliminate the data points affected by noise. The interaction between the user and the computer program is facilitated by a friendly graphical user interface.
We validated the method by using it to model two bi-dimensional mathematical function: a linear one and a nonlinear one. As an application of the method we use it to model a circuit function for a simple operational transconductance amplifier..

Key words: *data set, noise cancellation, fuzzy model, analog circuit function*

I. INTRODUCTION

The problem of estimating an unknown function from samples (i.e. function approximation from a finite number of data points) has been and is still a fundamental issue in a variety of scientific and engineering fields [1]. According with [2], approximation techniques for multidimensional functions are informally classified into five groups: sectionwise piecewise polynomial functions, canonical picewise linear representations, neuro-fuzzy interpolation, radial basis function and multilayers of nested sigmoids. The 3rd group exploits the interpolation capabilities of fuzzy inference.

Tagaki-Sugeno fuzzy systems are attractive for modeling complex functions because they are universal aproximators [3], [4], [5] and can model any nonlinear, multivariable function. The modeling approach used to build fuzzy models is similar to many system identification techniques. In general this type of modeling works well if the training data is fully representative of the features of the functions intended to be modeled

In this paper we propose a method to automatically find and eliminate the data points affected by noise. After that we use this noise free data set to build the final fuzzy model. As an application of the method we use it to model a circuit function for a simple operational transconductance amplifier.

The remainder of the paper is organized as follows. Section II presents the overview of the method. Section III shows the implementation of the method. We validate the method by using it to model two bi-dimensional mathematical functions: a linear one and a nonlinear one. In section IV we present an application in the field of analog circuits function modeling. Finally we conclude our paper in Section V.

II. OVERVIEW OF THE METHOD

Due to the fact that Takagi-Sugeno fuzzy systems can accurately approximate any complex multivariable functions, we have selected this class of models to automatically build computationally inexpensive models based on a data set.

A Takagi-Sugeno fuzzy systems is a particular case of fuzzy system because of its output membership functions that can be only linear or constant. The more general first-order Takagi-Sugeno system has rules of the form:

if x is A and y is B then $z = p \cdot q + q \cdot y + r$

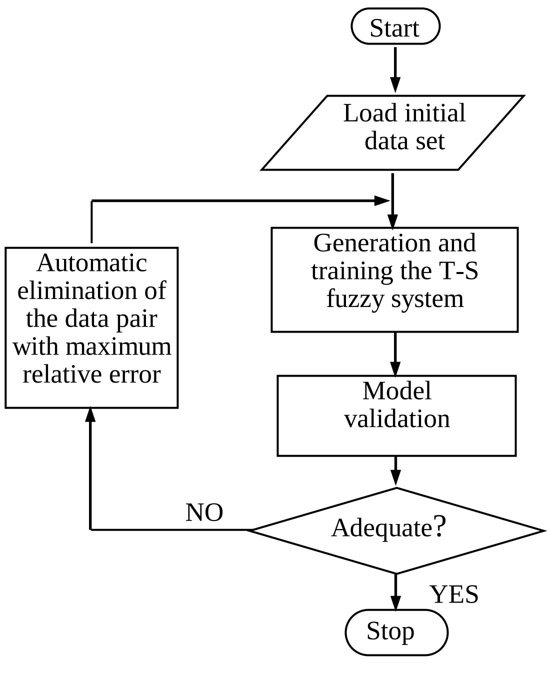

Figure1. Block diagram

where *A* and *B* are fuzzy sets in the antecedent while *p, q* and *r* are all constants. The easiest way to visualise the first-order system is to think of each rule as defining the location of a "moving singleton".

According to [6], [7], [5] for building fuzzy models we need a set of numerical data. To obtain accurate models a large number of data pairs is requested, that should uniformly cover the function domain and include the function characteristics, as much as possible. In Figure 1 the block diagram of the proposed fuzzy modeling method is shown.

With the initial data set we will generate and train a Takagi-Sugeno fuzzy system as a model for our function. The generation of initial fuzzy model uses a fuzzy subtractive clustering, the resulting clusters being used to extract the set of fuzzy sets and rules. The number of clusters (the same with the number of rules) is determined by a variable "radii" which specifies a cluster center's range of influence in each of the data dimensions. If we have a small number of fuzzy rules the system cannot approximate the function very well and if we have a big number of rules, the number of model coefficients, which need to be tuned, is very big and will need a greater number of data pairs. Therefore we have to find an optimum number of rules to build the circuit functions models. The range for radii is [0, 1]; a value close to 0 provides a large numbers of clusters, while a value close to 1 provides a small numbers of clusters. This initial fuzzy model is then trained with Adaptive-Network-based Fuzzy Inference Systems (anfis), which is the major training routine for Sugeno-type fuzzy inference systems. Anfis uses a hybrid learning algorithm to identify parameters of Sugeno-type fuzzy inference systems. It applies a combination of the least-squares method and the backpropagation gradient descend method for training membership function parameters to emulate a given training data set. The fuzzy inference system is trained during a number of epochs specified by the user. More details about anfis can be found for example in [4], [8].

Then we perform a model validation. In order to see the model accuracy and to see if the function values from the data set were affected by noise, we calculate the relative error between the function values from the data set and the function values computed with the fuzzy model, resulting an error vector.

The user sets a maximum acceptable value for the relative error. If this acceptable value is bigger than the maximum value in the error vector, than the model is considered adequate. Otherwise the algorithm performs the action to eliminate the noisy data. In each iteration we eliminate a data pair corresponding to the maximum relative error. With the new set of training data we generate, train and validate a new fuzzy model. The process stops if all the error vector values are smaller than the maximum accepted value, which means that the algorithm has eliminated successfully the noisy data and we have achieved our goal to obtain an accurate model. The process also stops if the actual number of data pairs is 25% smaller than the initial number of pairs. In this case we consider that the training data is inconsistent meaning that the initial data are very noisy or do not capture enough representative features of the function.

III. IMPLEMENTATION AND VALIDATION

We implemented our algorithm in Matlab and build a friendly graphical user interface as a communication bridge between the software and the user (Figure 2).

In order to see how our method works, we use it to model two mathematical functions: a linear one f(x,y)=x+y and a nonlinear one, $f(x,y)=e^x+x/y$. To test our algorithm we intentionally altered the value of the linear function in 7 data pairs and the value of the nonlinear function in 10 data pairs, as one can see in Table 1 and respectively in Tabel 2.

To build our fuzzy models we used for the linear function 300 data pairs and for the nonlinear function 500 data pairs.

We can see that even the initial fuzzy modes, built with noisy data, generates function values closed to the correct ones, the relative errors being in the range: (6.31%; 29.66%) while the relative errors in the noisy pairs were (22.63%; 253.6%) for the linear function. For the nonlinear function the relative errors in the noisy pairs were in the range of (27.7%; 900%) and with the initial fuzzy model the range of the relative errors was (12.5%; 69.8%). This is due to the interpolation capabilities of the fuzzy models.

After 11 respective 24 iterations the algorithm has eliminated all the noisy data pairs from both training sets. The finals fuzzy models provide more precise values of the functions, the relative errors being reduced with at least two

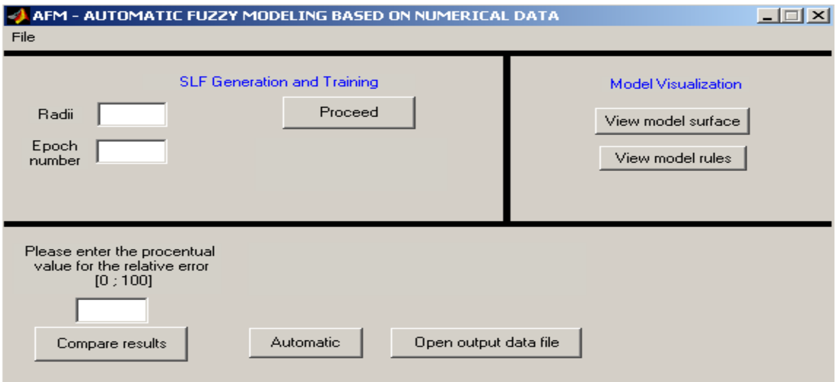

Figure 2 Graphical user interface

Table 1 Data for f(x,y)=x+y

Data Point		Correct Value	Noise Affected	Computed with fuzzy model	
				With noise	After noise cancellatio
1.	Function value	50.9	180	66	50.85
	Error [%]		253.6	29.66	0.098
2.	Function value	64	23	52.32	64.03
	Error [%]		64	18.25	0.046
3.	Function value	47.7	22.3	50.71	47.7
	Error [%]		53.24	6.31	0
4.	Function value	99	524	79.2	99.24
	Error [%]		429	20	0.242
5.	Function value	102.5	79.3	111.2	102.4
	Error [%]		22.63	8.48	0.097
6.	Function value	87.2	22.5	74.11	87.23
	Error [%]		74.19	15.01	0.034
7.	Function value	39.5	80	35.2	39.5
	Error [%]		102.5	10.8	0

Table 2. Data for $f(x) = e^x + \frac{x}{y}$

Data Point		Correct Value	Noise Affected	Computed with fuzzy model	
				With noise	After noise cancellation
1.	Function value	178	478	288	181
	Error [%]		168	61.7	1.68
2.	Function value	13.3	93	40.2	13.01
	Error [%]		599	202	2.18
3.	Function value	105	155	129	109
	Error [%]		47.6	22.8	3.8
4.	Function value	144	104	126	138.5
	Error [%]		27.7	12.5	3.81
5.	Function value	15.9	45.9	27	16.1
	Error [%]		188.9	69.8	1.25
6.	Function value	45	85	57	47.3
	Error [%]		88.8	26.6	5.1
7.	Function value	8.1	81	13.4	8.25
	Error [%]		900	65.4	1.85
8.	Function value	5.6	15.6	8.2	5.5
	Error [%]		178.5	46.42	1.78
9.	Function value	52.96	5.29	38	54.1
	Error [%]		90	28.2	2.15
10	Function value	108	198	127	109.3
	Error [%]		83.3	17.5	1.2

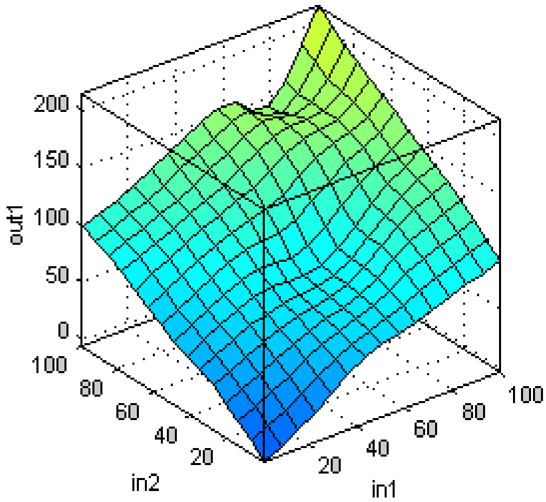

Figure 3.Fuzzy model surface for the linear mathematical function affected by noise.

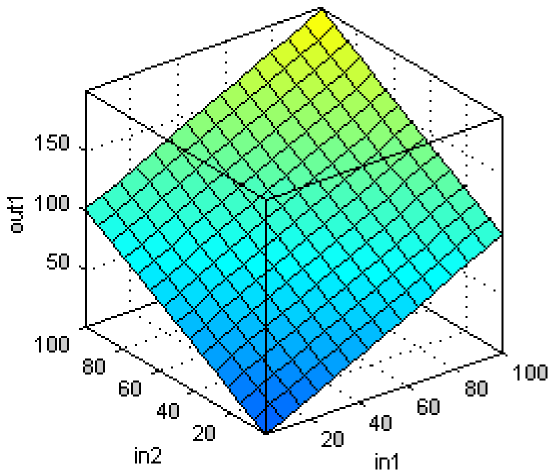

Figure4 .Fuzzy model surface for the linear mathematical function after noise cancellation.

magnitude orders. The maximum relative error for the final fuzzy model is decreased to 0.242% compared with 6.31% in the initial fuzzy model for the linear function and to 5.1% from 12.5% for the nonlinear one.

In Figure 3, Figure 4, Figure 5 and Figure 6 the functions values obtained with fuzzy models affected by noise and after noise cancellation are shown. It can be easily seen that after the noise cancellation (Figure 4 and 6) the fuzzy model surfaces are smooth.

A measure of the accuracy of the fuzzy model is the normalize root mean square error (NRMSE). [1], [9]:

$$NRMSE = \frac{RMSE}{\sigma} \quad (1)$$

where σ is the standard deviation and RMSE is the root mean square error:

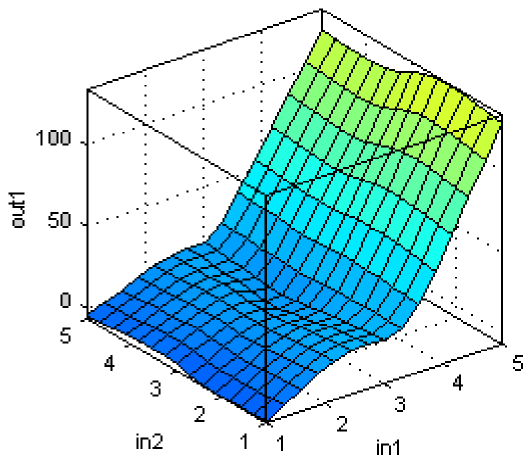

Figure5 .Fuzzy model surface for the nonlinear mathematical function affected by noise

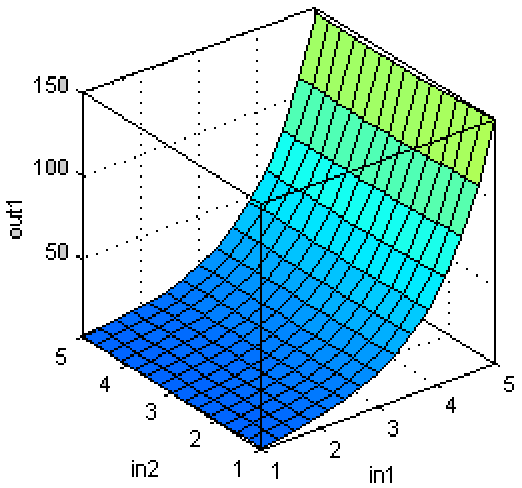

Figure 6.Fuzzy model surface for the nonlinear mathematical function after the noise cancellation

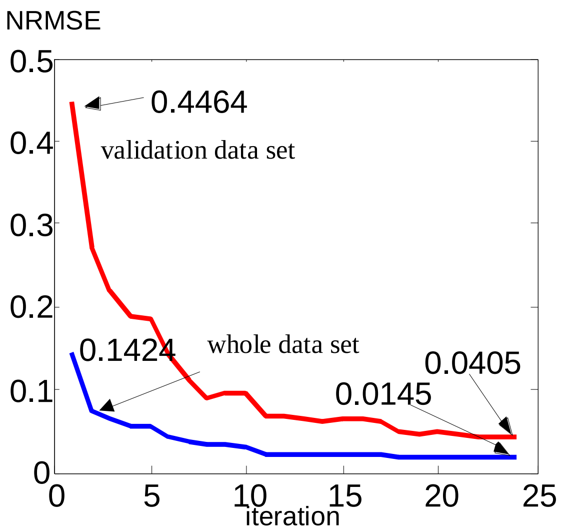

Figure 7 Evolution of NRMSE for the nonlinear function

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(P_i - P_{io}\right)^2} \quad (2)$$

where n is the dimension of the data set, P_i is the function value obtained with the fuzzy model for the i-th data pair and P_{io} is the function value of the i-th pair from the data set.

For the nonlinear function we considered a validation dataset with 10 data pairs, the same with the ones initially affected by noise, but with correct values for the function.

The evolution of NRMSE during the noise elimination algorithm is shown in Figure 7, with light line for this validation set. We can see a rapid improvement of the values computed with fuzzy models after each iteration (from NRMSE=0.4464 for the initial model down to NRMSE=0.0405 for the final model after noise cancellation).

The evolution of the NRMSE for the whole data set (values from fuzzy models compared with correct values) is also represented in Figure 7 with a dark line. The accuracy of the fuzzy models is improved iteration by iteration from 0.1424 initial value of NRMSE down to 0.0145 final value.

A small disadvantage of our method is that the automatic elimination of noisy data pairs have a collateral effect. Other good data points are also eliminated (7 good pairs for the linear mathematical function and 14 good pairs for the nonlinear mathematical), as it is shown in Table 3.

Table 3

Function	Number of data pairs	Noisy data pairs	Eliminated data pairs
Mathematical two variables linear function f = x + y	300	4,9,15,21, 22,35,40	**35, 4,** 123, **15, 40,** 56, **22,** 34, **9,** 164, **21**
Mathematical two variables nonlinear function $f = e^x + x/y$	500	4, 27, 47, 48, 67, 127, 190, 254, 390, 465	**4,** 105, 8**, 67, 27,** 34 **47,** 119, 2, **254,** 318, 221, **27,** 157 239 **190, 8,** 268, 297, **465,** 88, 306,167, **390**

IV. APPLICATION IN CIRCUIT FUNCTIONS MODELING

In this section we apply our method to build fuzzy models in the field of analog circuit functions

We used our algorithm for the voltage gain (avo) of the simple operational transconductance amplifier(sota) (Figure 8).

The design parameters of the circuit are the dimensions of the transistors and the bias current I_b. Happily the parameter numbers will decrease after a simple analysis of the circuit. The input transistors Q_1 and Q_2 must be identical, therefore $(W/L)_1 = (W/L)_2$ resulting the first parameter $(W/L)_{12}$. The transistors Q_3 and Q_4 which form the active load must be paired, resulting $(W/L)_3 = (W/L)_4$, so our second parameter will be $(W/L)_{34}$. For the current mirror formed by Q_5 and Q_6 we consider the current (I_b) equal trough both transistors so $(W/L)_5 = (W/L)_6$. In order to keep an minimal area, we have taken W=L so we obtained our

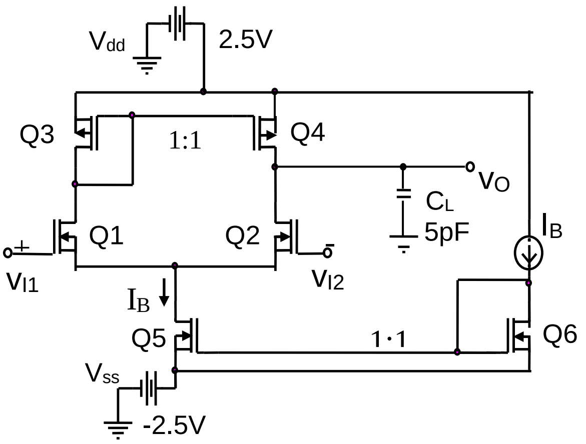

Figure 8.Simple operational transconductance amplifier

third parameter $(WL)_{56}$ with $W_5=L_5=W_6=L_6$. The fourth and final parameter is I_b.

The range of the parameters should be choused so that regardless the combination of the parameters values, the circuit always operates as an amplifier (all the transistors stay in the active region). In order to assure a high value for the voltage gain, the transistors Q_1 and Q_2 should be biased with a small overdrive voltage $V_{GS12} - V_{Pn} \approx 0.2V$ [10]. To stay in the active region $V_{DS12} > V_{GS12} - V_{Pn} = V_{DS12\,sat}$ and $V_{DS12} > 0{,}2$ V must be fulfilled. For Q_5 and Q_6 we have $V_{DS56} > V_{GS56} - V_{Pn} = V_{DS56\,sat}$ and taking into consideration their connection $V_{DS56} = V_{GS56} > V_{DS56\,sat}$. For the same reason Q_3 and Q_4 will always stay in the active region. If the transistors work at higher overdrive voltages the matching is better.

It is recommended for Q_3, Q_4 and Q_5, Q_6 to fulfill the relation:

$$\left|V_{GS} - V_p\right| \approx [0.5; 0.7] \tag{3}$$

The modeling parameters used in simulation are for the 0.25µm technology.

Due to the fact that parameters $(W/L)_{34}$ and $(W/L)_{56}$ have a very low influence on the voltage gain we keep only the parameters I_b and $(W/L)_{12}$. So the resulting values for our parameters are:

$$I_b \in [20; 70]\mu A;$$
$$(W/L)_{12} \in [1; 8] \text{ with } L_1 = L_2 = 0.5\mu m$$
$$(W/L)_{34} = 5 \text{ fixed};$$
$$(W/L)_{56} = 1 \text{ fixed}$$

The voltage gain avo is:

$$avo = \frac{v_o}{v_{i1} - v_{i2}} \tag{5}$$

The simplified mathematical expression for the voltage gain can be taken from the literature [10],[11], and it is:

$$avo = 2V_E L \sqrt{\frac{I}{I_B} K_n \left(\frac{W}{L}\right)_1} \tag{5}$$

where K_n is the transconductance parameters and $V_E L$ is:

$$V_E L = \frac{V_{EN} \cdot L_1 \cdot V_{EP} \cdot L_3}{V_{EN} \cdot L_1 + V_{EP} \cdot L_3} \tag{6}$$

with V_{EN} and V_{EP}, Early voltages for n channel and p channel MOS transistors.

More information about sota fuzzy modeling can be found in [12], [13]

For obtaining uniformly distributed points we used the Latin Hypercube Sample (LHS) technique [2] to generate the parameters pairs. Also with this technique we can cover the entire range of the function values and capture all its relevant features.

The parameters values and the resulting function values compose the input data set.

To see how our algoritm works, we intentionally altered the avo value for 5 data pairs as they would be like if they were noise affected, as one can see in Table 4. The avo values provided by fuzzy models before and after noise cancellation are also presented in Table 4. As we expected after the noise cancellation avo values are very close to the correct ones.

The fuzzy model for avo was built using 850 data pairs (parameters-function values) obtained from Pspice simulation.

Table 4

Data Point		Correct Value	Noise Affected	Computed with fuzzy model	
				With noise	After noise cancellation
1.	Function value	31.04	91.04	33.6	30.10
	Error [%]		193	8.24	2.93
2.	Function value	38.54	18.54	36.1	37.41
	Error [%]		51.8	6.33	2.93
3.	Function value	28.32	88.32	31.9	29.05
	Error [%]		211	12.64	2.57
4.	Function value	44.57	24.57	54.7	46.03
	Error [%]		44.8	22.72	3.27
5.	Function value	31.17	71.17	34.2	32.01
	Error [%]		128	9.72	2.69

In Figure 8 and Figure 9 the fuzzy models affected by noise and after noise cancellation are shown

One can see that the final surface (Figure 9) is more smoothly that the initial one, being in accordance with the function dependence on the parameters.

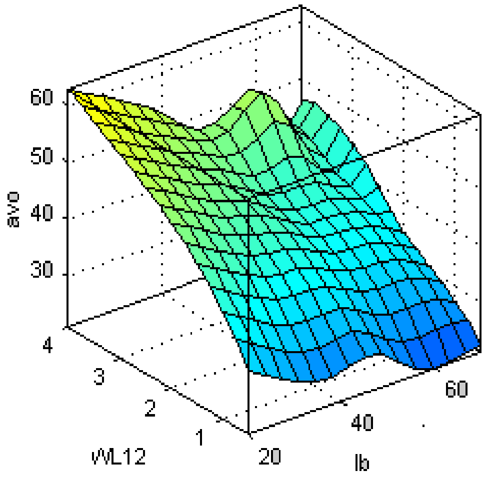

Figure 8.Fuzzy model surface for the analog circuit function affected by noise

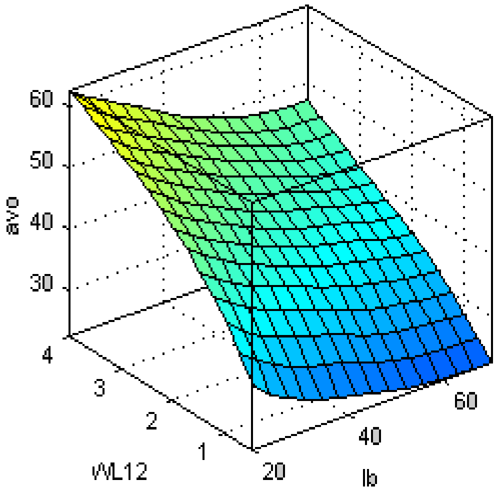

Figure 9 .Fuzzy model surface for the analog circuit function after the noise cancellation

V. CONCLUSION

An automatic method to build fuzzy models, by noise cancellation was presented. The results obtained in modeling three kind of functions (a mathematical linear one, a mathematical nonlinear one and a nonlinear circuit function) prove the efficiency and utility of the proposed method. The noisy data were successfully eliminated and the fuzzy models obtained after noise cancellation have a high accuracy degree, the maximum relative error in the final fuzzy models being 0.242% for the mathematical linear function and 3.27% for the nonlinear circuit function.

In the noise cancellation process a collateral effect was noticed. A small number of good data points were also eliminated besides the noisy ones. The number of good data pairs eliminated is insignificant because the data pairs were reduced with only 4% for the mathematical linear function, with 4.8% for the nonlinear mathematical function and with 1.4% for the nonlinear circuit function.

The applicability domain of the proposed method is practically unlimited; it can be used to model any kind of multivariable, complex function. The only requirement is to provide enough data pairs to capture all the relevant characteristics of the function.

REFERENCES

[1] Pomares, H., et all, “A Systematic Approach to a Self-Generating Fuzzy Rule-Table for Function Application”, *IEEE Transaction on System, Man, and Cybernetics – Part B: Cybernetics*, Vol. 30, No. 3, June, 2000, pp. 431-447;

[2] Chen, W.-K., Editor, *Synthesis and Design of Nonlinear Circuits*, in The Circuit and Filters Handbook, CRC Press, 1995 pp935-945;

[3] Mendel, J.M.,”Fuzzy Logic Systems for Engineering: A Tutorial”, *Proc. of the IEEE*; vol.83 No. 3, March 1995;

[4] Godjevac, Jelena, *Neuro-Fuzzy Controllers. Design and Application*, Presses politechniques et universitaries romandes, Lausanne; 1997;

[5]Torralba, A., Chavez, J., Franquelo, L.G, “A collection of Fuzzy Logic-Based Tools for the Automated Design, Modelling and Test of Analog Circuits”, *IEEE Micro*, 1996,vol 16, no 4, pp 61-68;

[6] *** , *Fuzzy Logic Toolbox, User’s Guide*, Fourth printing, Revised for Matlab 6.0 (R12), The Math Works, Inc; 2000

[7] Gieger, R., L., Allen P., E., Strader N., R., *VLSI Design Techniques for Analog and Digital Circuits*, McGraw-Hill Publishing Company. 1990

[8] Jang, R. J.-S., “ANFIS, Adaptive-Network-Based Fuzzy Inference System,” *IEEE Transaction on Sstem, Man, and Cybernetics*, Vol. 23, No. 3, pp. 665-685,1993

[9] Oltean, G., *Fuzzy Mathematics in Analysis and Design os Some Electronic Circuits*, Cluj Napoca, 22 iunie, 2002. –PhDThesis

[10] Baker, R.J., Li, H.W., Boyce, D.E., *CMOS-Circuit Design, Layout and Simulation*, IEEE Press, Pistcataway; 1998

[11] Hershenson, M.D.M., Boyd, S.P., Lee, T.H., (), “GPCAD: A Tool for CMOS Op-Amp Synthesis”, *International Conference on CAD 98*; 1998

[12] Crasi, M., Oltean, G., "Fuzzy Modeling of Circuit Functions for Simple Operational Transconductance Amplifier", *IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics*, A&QT-R 2002 JUNIOR Branch, May 23-25, 2002, Cluj-Napoca, Romania, Tome III, pp. 11-16..

[13] Oltean, G., Miron, C., Crasi, M., Carlugea, Mihaela, “Evaluation of the Analog Circuits Performances using Fuzzy Models”, *Proc. of the Symposium on Electronics and Telecommunications*, Etc.2002, Timisoara, Sept. 2002, Vol I, pag. 12-17.