Fuzzy Techniques in Analog Circuit Design

GABRIEL OLTEAN
Bases of Electronics Department
Technical University of Cluj-Napoca
C. Daicoviciu Street, 15, 400020, Cluj Napoca,
ROMANIA,
Gabriel.Oltean@bel.utcluj.ro

Abstract: - The aim of his paper is to present some applications of fuzzy techniques in the optimization-based design of analog circuits. Our approach turns into profit the advantages offered by different fuzzy techniques. Fuzzy systems or fuzzy sets are involved in every algorithm phase. The optimization problem formulation is accomplished in a flexible manner using fuzzy sets to define fuzzy optimization objectives. Also, the initial guess of design parameters is based on matching degrees determined with fuzzy sets. The optimization engines use fuzzy systems to compute the coefficients to modify the design parameters in each iteration. In order to reduce the time spent for circuit performance evaluation, we use a fuzzy system to model each circuit performance. Two computer-aided design tools, at the cell level are developed in Matlab. A large collection of experimental results proves the validity of our approach. Different analog modules as simple transconductance operational amplifier, Miller operational transconductance amplifier, and common-emitter stage were designed for several sets of design requirements with very good results.

Key-words: Analog design, Multiobjective optimization, Fuzzy sets, Fuzzy systems, Unfulfillment degree.

1. Introduction

Actual trends in VLSI technology are towards integration of mixed analog-digital circuits as a complete system-on-a-chip. Most of the knowledge-intensive and challenging design effort spent in such systems design is due to the analog building blocks [1]. Analog design has been traditionally a difficult discipline of IC design. In circuit design optimization, a circuit and its performance specifications are given and the goal is to automatically determine the device sizes in order to meet the given performance specifications while minimizing a cost function, such as a weighted sum of the active area or power dissipation [2]. This is a difficult and critical step for several reasons: 1) most analog circuits require a custom optimized design; 2) the design problem is typically underconstrained with many degrees of freedom; and 3) it is common that many (often conflicting) performance requirements must to be taken into account, and tradeoffs must be made that satisfy the designer [3].
Consequently, the development of CAD tools at the cell-level, that automate and speed up the design process of analog portions of circuits and systems remains as an active research area in both industry and academia [1].
Fuzzy techniques have been successfully applied in a variety of fields such as automatic control, data classification, decision analysis, expert systems, computer vision, multi-criteria evaluation, modeling, optimization, etc.
Works showing the possibility of application of fuzzy logic in computer aided design of electronic circuits started to appear in late 1980s and early 1990s. An argument for fuzzy logic application in CAD is derived from the nature of the algorithm used for solving design problems. The majority of algorithms for design synthesis use heuristics that are based on human knowledge acquired through experience and understanding of problems. The natural language, a fuzzy logic language is the most convenient way to express such knowledge. Linguistic descriptions are usually given in fuzzy terms not only because this is the most common form of representation of human knowledge, but also because our knowledge about many aspects of the design is fuzzy [4]. Linguistic information, while not precise, represents an important source of knowledge. Another important source of knowledge is numerical data. Fuzzy logic systems are appropriate in such situations because they are able to deal simultaneously with both types of information: linguistic and numerical.
Also, fuzzy systems being universal approximators can model any nonlinear functions of arbitrary complexity [5], [6]. This is very useful in modeling complex circuit functions of high accuracy at low cost, necessary in performances evaluation.
The objective of our research is to turn into good account the advantages of fuzzy techniques in the optimization-based analog circuit design field. We developed optimization algorithms that use fuzzy

approaches in all their phases. Two CAD tools at the cell level, with friendly graphical user interfaces are developed in Matlab.

The reminder of the paper is organized as follows. We begin in Section 2 with an overview of the optimization algorithm used in the design optimization of analog modules. Section 3 focuses on fuzzy sets utilization in the formulation of the multiobjective optimization problem. The method to obtain a good initial solution, based on matching degree, is presented in Section 4. Evaluation engine, based on fuzzy models of circuit performances is the subject of Section 5. The detailed procedure to build such fuzzy models and some results for a simple operational transconductance amplifier are also included. The heart of the algorithm, the optimization engine is address in Section 6. Two fuzzy system-based engines (global gradients fuzzy optimization and local gradients fuzzy optimization) are explained in some details here. Section 7 discusses the implementation of our fuzzy-based CAD tools and some results designing three analog modules. In the end, in Section 8, some conclusions are drawn.

2. Optimization Algorithm

Design optimization of an electronic circuit is a technique used to find the design parameter values (length and width of MOS transistors, bias current, capacitor values etc) in such a way that the final circuit performances (dc gain, gain-bandwidth, slew rate, phase margin etc.) meet as close as possible the design requirements.

In contrast with the circuit analysis, which is a direct problem, the circuit design is a reverse problem. The goal of the reverse problem is to determine a cause (values of the design parameters) that produces an arbitrarily specified effect. Due to the arbitrarily character of the specified effect it is possible for the design problem to have no solution. Even though a solution exists, generally it is not unique.

Finding design solutions is difficult due to concurrent requirements and complex nonlinear relations connecting the circuit performances on the design parameters. As well, the number of parameters and the number of design requirements (design equations) are different, so we have to deal with an over-determined or under-determined system of nonlinear equations.

There is no general design procedure independent of the circuit; also, there is no formal representation to connect the circuit functions on its structure in a consistent manner. The major obstacle consists in the peculiarity of the analog signals: the continuous domain of the signals' amplitude and their continuous time dependency. Hereby the analog circuit design is known like an iterative, multi-phase task that necessitates a large spectrum of knowledge and abilities of designers.

As stated in [7] there are two basic modalities to deal with the analog design: knowledge based approaches and optimization based approaches. In the present paper we are centered on the last one. The optimization algorithm begins with the formulation of the optimization objectives and optimization problem, followed by the initialization of the design parameters. During iterations an evaluation engine computes the actual circuit performances based on the actual design parameter values. If the objectives are fulfilled, the solution consists in the set (or sets – in the case of a real multiobjective optimization) of the actual design parameter values and the algorithm is stopped. If not, new design parameter values are to be computed by the optimization engine and the optimization loop is covered once again.

The novelty introduced in this paper is the utilization of different fuzzy techniques in every phase of the optimization algorithm, as it is shown in Fig. 1. Fuzzy sets are involved in the formulation of the optimization objectives and in the initial guess of the design parameters, while some fuzzy

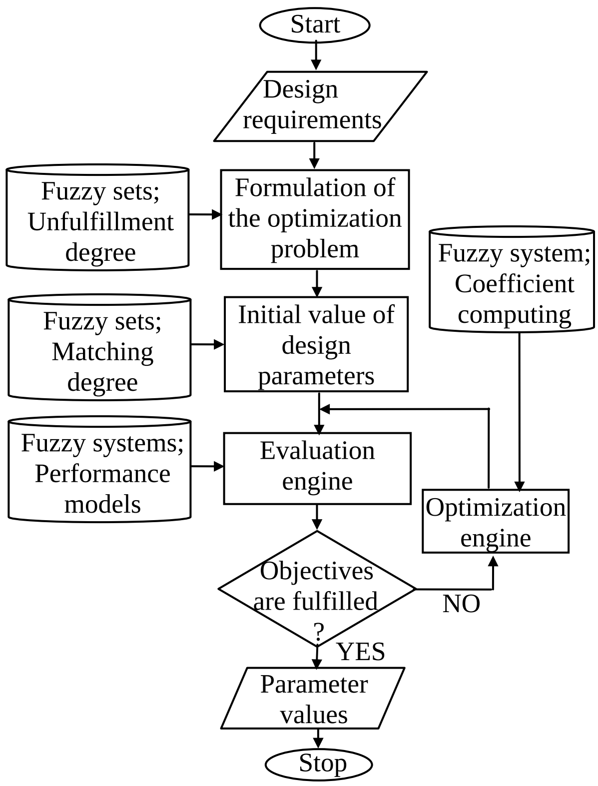

Fig.1 Fuzzy optimization algorithm

systems address the performance evaluation issues (evaluation engine) and new parameter values computation (optimization engine).

3. Optimization Objectives Formulation

To formulate the design objectives for a real design is not always a simple task. Often, it is not clear what precise values to give to each objective. In fact design objectives are often better expressed in real world terms than in precise numbers. The designers usually accept a certain degree of fulfillment of the design objectives. This comes mainly from then reality of fabrication process where accurate control of every step of the fabrication process becomes more and more difficult and provides differences between computed values and fabricated values [8].

The fuzzy techniques used to define the optimization objectives suppose the fuzzification of the requirements, getting this way the possibility to consider different degrees for requirement achievements and acceptability degrees for a particular solution. One or two fuzzy sets are associated with each requirement. Their membership functions represent the corresponding fuzzy objective functions. One approach [9] - [14] is to consider that the membership degree μ represents the degree of fulfillment of the fuzzy objective. A value μ=1 means that the objective is fully satisfied, while a value μ=0 means that the objective is not satisfied at all. This method suffers a disadvantage in the case of equality requirement. No information is provided regarding the relation between the requirement and the actual performance, as the performance is greater than the requirement and it should be decreased, or the performance is smaller than the requirement and it should be increased.

This disadvantage can be overcome using the method proposed by the author [15]-[18]. The membership degree μ represents the error degree in the fulfillment of the fuzzy objective. A value μ=1 means the objective is not satisfied at all, while a value μ=0 means the objective is fully satisfied.

As an example, the requirements "greater or equal" $f_k(x) \geq f_k^r$ and "equal" $f_k(x) = f_k^r$ have the corresponding fuzzy objective functions presented in Fig 2. where:

x – the vector of the design parameters;

f_k - the k^{th} performance function;

f_k^r - the k^{th} requirements;

x^* - the current value of the design parameters vector.

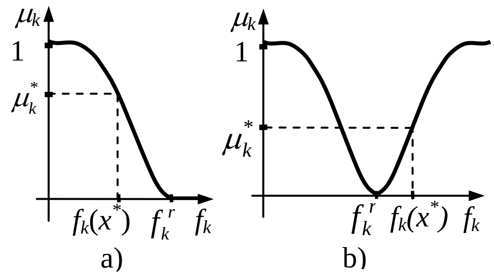

Fig. 2 Fuzzy objective functions:
a) $f_k \geq f_k^r$; b) $f_k = f_k^r$

The fuzzy objective functions are:

$$\mu_k\left(f_k(x)\right):\ D_{f_k} \rightarrow [0,1] \qquad (1)$$

where D_{f_k} is the range of possible values for $f_k(x)$. $\mu_k\left(f_k(x)\right)$ indicates the error degree in accomplishing the k^{th} requirement, so we will call it "unfulfillment degree" (*UD*). A value μ_k=0 means full achievement of fuzzy objective, while a value μ_k=1 means that the fuzzy objective is not achieved at all. This occurs when $f_k(x)$ takes an unacceptable value. Fig 2 shows the corresponding value of the unfulfillment degree μ_k^* for the current value of the variables vector x^*. The optimization problem formulation became:

Find x that
minimizes $\{\mu_1(f_1(x)), \mu_2(f_2(x)),\ldots, \mu_n(f_n(x))\}$ (2)

where n is the number of requirements.

To solve such a multi-objective optimization problem there are two possibilities. The first one is to convert the problem into a single objective optimization problem, by combining all the individual objectives into a single objective function (for example using a weighted sum). The second approach involves a real multiobjective optimization method, as it is discussed in Section 6.

Fuzzy objectives have some advantages compared with the crisp ones. Fuzzy objectives provide an interface between the real world design problems and the mathematical formulation supported by most optimization algorithm. Tradeoffs are handled by manipulating the shape of the membership functions that reflect the fulfillment or violation of the design requirements.

4. Initial Design Parameters

Without a well-chosen starting point an optimization run may converge very slowly or converge to a local minimum with a significantly worse performance

than the circuit's best capability [19].
Different methods to obtain initial solutions are presented in the literature. In [10], [20], and [21] an experienced user is the one that provides a good initial guess. Approximate analytic design equations are used in [10], [11], and [22] while some randomly generated initial solution are preferred in [13], [19], [23], and [24].
The author proposed a fuzzy based method for the initial solution [18] and [25]. It is based on a selection of the initial solutions from a larger set of design parameter vectors, previously generated using Latin Hypercube Sample technique. The selection criterion is the matching degree of the corresponding initial performances with the design requirements. For every requirement a fuzzy set is used to define the matching degree, as for example in Fig. 3 for an "equal" type requirement and for a "greater or equal" type requirement.

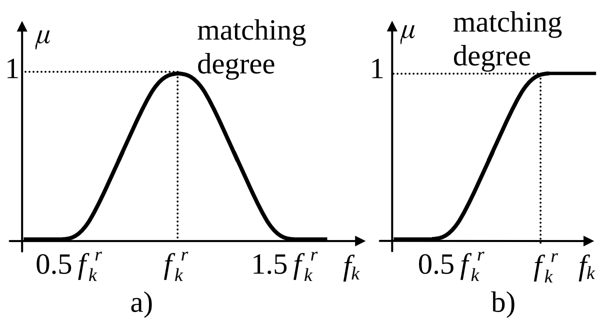

Fig. 3. Fuzzy sets defining the matching degree: a) $f_k = f_k^r$; b) $f_k \geq f_k^r$

For every set of initial performances an arithmetic mean of the individual matching degrees is computed. The *m* initial solutions are sorted in descending order of the mean matching degree and the first *n* are chosen as initial candidate solutions.
This method is appropriate for optimization methods based on a population of solutions. If a conventional method is used, the initial solution with highest matching degree can be selected.
The fuzzy method presents positive aspects: no necessity for an experienced designer, good quality of the initial solution and no user intervention.

5. Evaluation Engine

The design process of an electronic circuit is an iterative process (see Fig.1) that requires a large number of performance evaluations. Analog circuits are difficult and time-consuming for a proper evaluation.
Performance modeling of an analog integrated circuit involves the representation of a circuit behavior in terms of its design parameters (component sizes, bias currents and voltages).
Even in the case of basic characteristics of a simple circuit (amplifier gain, gain-bandwidth, slew rate etc) the performance in question can be a complex function of many parameters. In a realistic case, a performance model will be in general a non-linear function over a high dimensional space of circuit parameters [26].
An accurate estimation of the circuit performances requires the use of complex models (for example SPICE simulation) leading to an excessively large computation time. One way to reduce the computation time is to use more simple models of circuits and devices. In order to satisfy both main requirements (accuracy and speed), many researches proposed several methods to evaluate circuit performances.
Simple analytical models are used in [11], [20], [21], [27], and [28], while some version of polynomial and monomial models are involved in [29], [30] and [2]. A lot of automatic design tools include a Spice-like simulator to accurately compute the performances [31] - [35] and [2]. Least-square support vector machine are also involved in [26], [36], and [37].
Fuzzy systems are very useful to model the circuit performances because they imply just a few simple mathematical operations and can model any complex, multivariable and nonlinear function at any level of accuracy. Such fuzzy models are used in [1], [38], and [39].
The author synthesized a method to build such models and used it for some analog modules [16], [18], [40]. These models can be automatically built up using an input-output data set.
Each circuit performance is modeled by a first order Takagi-Sugeno system, having the circuit parameters as inputs and the performance function as output (see Fig. 4.).

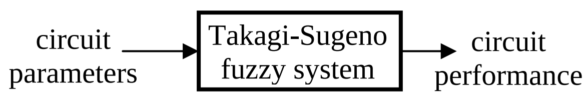

Fig. 4 Fuzzy model of a circuit performance

The full modeling procedure is explained in Fig. 5. The procedure starts with the analog circuit whose performances are to be modeled.
The ranges of the parameter values are established so that irrespective of the parameter values combinations, the circuit must operate in the desired region. For example, in an amplifier the transistors should be maintained in their active regions.
The parameter set (the combination of the parameter

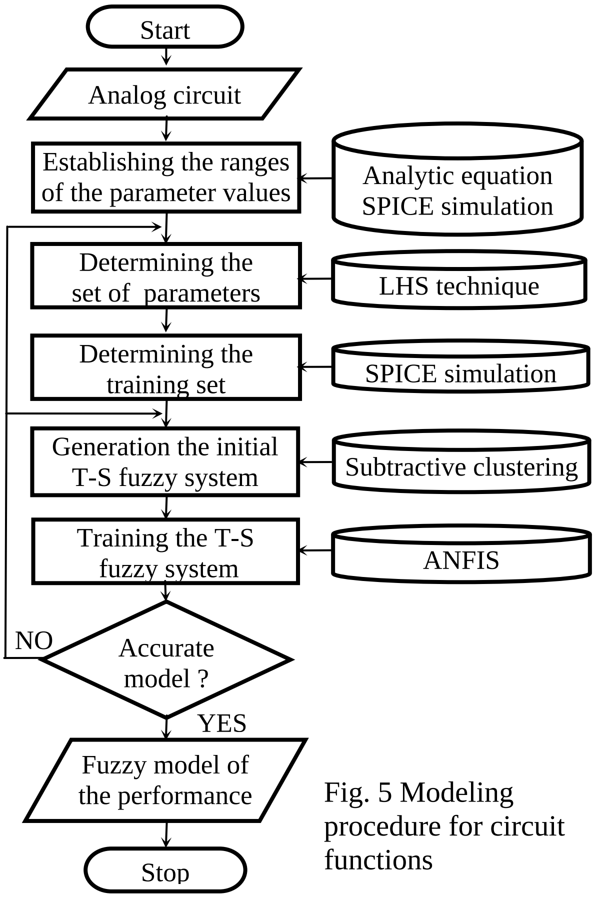

Fig. 5 Modeling procedure for circuit functions

values) should be chosen to be representative for the function to be modeled (covers the space of the parameters and embeds all the specific characteristics of the function).

For each input vector (one combination of the parameter values), the function value has to be found, in our case by SPICE simulation. All input vectors, together with their corresponding function values compose the data sets. Two data sets: training set and checking set are generated.

The training set is then subdued to a subtractive clustering procedure resulting an initial first order Takagi-Sugeno fuzzy system**.** Next, the initial fuzzy system is trained using ANFIS – Adaptive Neuro-Fuzzy Inference System and the training data set. The resulting fuzzy model is tested from the accuracy point of view. If the accuracy is to low, the procedure must be resumed by generating a new initial fuzzy system or even by determining new data sets. If the accuracy is acceptable the modeling procedure stops and provides the desired fuzzy model of that circuit function.

Using fuzzy models in the evaluation engine, the accuracy can be very good at a high speed. The accuracy is good, but not sufficient to pass the resulted design directly to an industrial manufacturing process, where an industrial strength simulator is used to validate the circuit. One idea [20] is to perform the optimization in two steps: first step uses non-Spice (fuzzy) models for speed, and the second step uses Spice models for accuracy. Because the starting point for the second phase is very close to the final solution only few iteration are necessary to carry out the optimization process, so the overall time will not increase too much. The main advantage of fuzzy models is that there are no restrictions in the kind of functions that can be modeled, as far as fuzzy systems are universal approximators. So, developing such models for circuit functions can be a very useful task in the field of analog design.

5.1. Fuzzy Models for SOTA

This paragraph presents some results regarding the fuzzy models obtained for the circuit functions in the case of a simple operational transconductance amplifier (SOTA) (see Fig. 6).

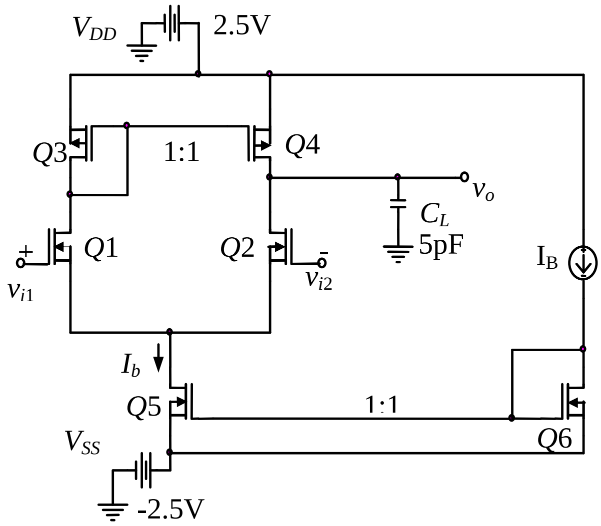

Fig. 6 Simple operational transconductance amplifier (SOTA).

The design parameters of the circuit are the dimensions of the transistors *(W/L)* and the bias current I_b. The input transistors Q_1 and Q_2 must be identical, therefore $(W/L)_1=(W/L)_2$ resulting the first parameter $W12=(W/L)_{12}$. The transistors Q_3 and Q_4 (active load) must be paired, resulting $(W/L)_3=(W/L)_4$, so our second parameter will be $W34=(W/L)_{34}$. For the current mirror, 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 a minimal area, we have taken W=L so we obtained our third parameter $W56=(W/L)_{56}$. The fourth and final parameter is I_b.

As performances, the important ones were considered: voltage gain A_{v0}, unity gain bandwidth GBW, phase margin PM and common mode rejection ratio $CMRR$.
Applying the previously described procedure, we built our fuzzy models for each circuit function with a set of 850 data pairs (700 training pairs and 150 checking pairs).

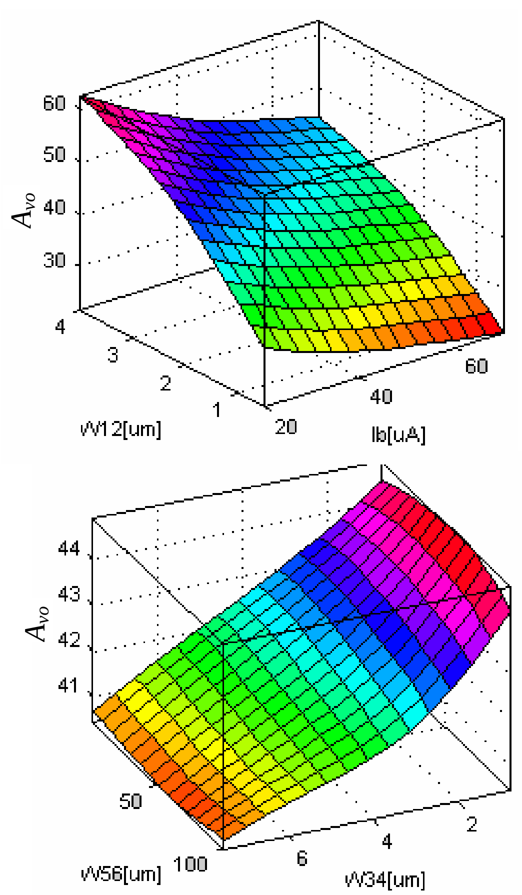

Fig. 7. The surface of the A_{vo} fuzzy model

The resulted surfaces of the A_{vo} fuzzy model are presented in Fig. 7. Because the function depends on four variables, two different plots appear, each of them representing the A_{vo} dependence on two variables: $I_b, W12$, (top), respectively $W34, W56$ (bottom). In each picture, the values of the other two parameters were set to the middle of their ranges. A_{vo} depends heavily on the $I_b, W12$ parameters, considering its quite large variation (from 20 to 63). A_{vo} increases with $W12$ and decreases with I_b, its maximum value being obtained for maximum $W12$ and minimum I_b. The A_{vo} dependence on the other two parameters, $W34, W56$ is reduced, the variation of the function lying in a narrower range (from 40.5 to 45).

To check the accuracy of the fuzzy model, we tested it against two sets of data, a test set (data that was not included neither in the training or checking sets) and a verification set (data that was included in the training set). As accuracy metric we used the absolute relative error, shown in Table 1. Mean and maximum values of this error are presented for all the performance functions. The mean errors are relatively small, confirming the good modeling accuracy for these multi-variable complex nonlinear functions.

Table1. Errors for fuzzy models

Circuit function	Data set	Relative error [%]	
		Mean	Maximum
A_{vo}	Test	0.83	1.01
	Verification	0.85	2.57
GBW	Test	3.07	8.53
	Verification	1.81	5.20
PM	Test	0.03	0.10
	Verification	0.02	0.11
$CMRR$	Test	3.04	8.74
	Verification	4.67	9.24

6. Optimization Engine

The optimization engine (the way to compute new parameter values) is the heart of the optimization algorithm. It should be chosen so that the optimization will converge to an optimal solution in a reduced number of iterations. This task is not an easy one due to complex relations between design parameters and circuit performances. The same parameter can affect more than one circuit performance at a time, so when a parameter is modified to improve a performance it can worsen another.
The literature abounds in optimization methods. Some classic (local) optimization methods involved in analog design automation are: steepest descent [19], [22] and [41], sequential quadratic programming [38], [42], Lagrange multiplier [22], conjugate gradient [20], feasible direction [10], or simplex [4]. Global optimization based on classical methods are also present in the field: simulated annealing [20], stochastic pattern search [24], geometric programming in convex form [29], [30], and [2]. Computational based method as genetic algorithm and evolutionary strategy [1], [31], [43], and [44] proved to be another class of efficient optimization methods.
The author proposed two fuzzy optimization methods based on fuzzy inference systems. The first one, GGFO (Global Gradients Fuzzy Optimization)

uses global qualitative dependencies (qualitative gradients) of the performance functions on the design parameters. The second optimization engine, LGFO (Local Gradients Fuzzy Optimization) is based on local quantitative gradients.

6.1 Global Gradients Fuzzy Optimization

The GGFO method proposes [15] and [36] a zero order Takagi-Sugeno fuzzy system for every design parameter to compute a coefficient (*coef*) to modify it (Fig. 8.).

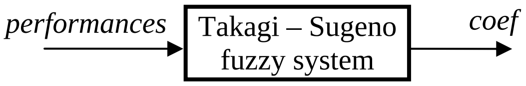

Fig. 8 Coefficient computing

The fuzzy sets for the inputs (*performances*) are the same with those used to formulate the optimization problem. The coefficient takes values in [-1; 1] and the corresponding parameters will be modified as follows:

$$param_{ij} = \begin{cases} \left(1+\eta_u coef_{ij}\right) param_{ij-1};\ coef_{ij} \geq 0 \\ \left(1+\eta_d coef_{ij}\right) param_{ij-1};\ coef_{ij} < 0 \end{cases}$$

where η_u and η_d are two dump factors with default values η_u =1, η_d =0.5, and i=1, 2, …, N is the index of the parameter and j=1, 2, …, M is the current iteration index.

The fuzzy rule base is build using the qualitative dependencies between design parameter and each circuit performances. For example let us consider the qualitative dependences of the f_1 and f_2 performance functions on the x_1 and x_2 parameters presented in the Table 2 (the arrows show the variation direction). The fuzzy system rules for $coef_2$ (the iteration index is omitted) are written based on Table 2:

Table 2. Qualitative dependence functions - parameters

Performances \ Parameters	x_1	x_2
f_1 ↗	↗	↘
f_2 ↗	↗	↗

R1: **If** f_1 is bigger or f_2 is smaller
then *coef2* is positive

R2: **If** f_1 is smaller or f_2 is bigger
then *coef2* is negative

This is a local multioptimization method so it can find an acceptable local optimal solution. It should be noticed that this approach holds only for monotone performance function in respect with every parameter. We can increase the chances to find the global optimum solution if we repeat several times the optimization procedure for different initial design parameter.

6.2 Local Gradients Fuzzy Optimization

The second fuzzy approach uses local quantitative gradient information (LGFO) [18], [45]. The optimization starts with the initial candidate solution. Each design parameter can affect more or less each objective function. The sign and the value to modify a certain design parameter take into account the unfulfillment degrees, the quantitative gradients and the relative importance of the involved variables in relations with the objectives.

The method acts as a human expert for a certain circuit performance:

- modifies more the parameter with greater importance, because it can really affect the performance, and the modification also depends on the unfulfillment degrees of the corresponding requirements.
- the parameter with lower importance is modified less or not at all, because its influence on circuit performance is insignificant.
- the final modification of a parameter is a weighted sum of the partial modification (imposed by every objective function).

Such human expert knowledge is captured and incorporated in the optimization engine by means of a fuzzy logic system. For every x_i parameter and every f_k function, a partial coefficient $partc_{i,k}$ to modify that parameter is computed in each iteration by a first order Takagi-Sugeno fuzzy system. The fuzzy rules are presented in Table 3 where, for example the 4th column and the 3rd row give the following fuzzy rule:

Table 3. Fuzzy rules for partial coefficient

importance \ *UD*	Z	S	M	L
Z	Z			
S		VS	S	M
M		S	M	L
L		S	L	VL

Z – Zero
VS – Very Small
S -Small
M – Medium
L – Large
VL – Very Large

If *UD* is Medium and *importance* is Small
then *partc* is Small.

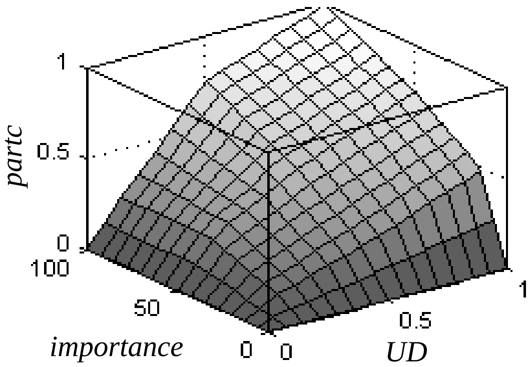

Fig. 9 Control surface for partial coefficient

The control surface generated by this fuzzy system is presented in Fig. 9. Finally, we should mention that the optimization method acts in an adaptive manner: when the *UD*s are great (towards 1), large coefficients to modify the parameters result. For small *UD*s result small coefficients to modify the parameter. Thus we can focus our search so that the solution converges to the optimal point. This method falls in the category of local optimization method.

In order to obtain a set of local Pareto optimal points for our multiobjective optimization problem, the method was developed to deal with multiple parallel search paths by means of a population of solutions, consisting of candidate solutions. During iteration, for every candidate solution, the actual performance, the *UD*s and new parameter values are computed. If the *UD*s for one candidate solution can not be decreased anymore, we have found a local Pareto optimal point and the future iterations will not visit this candidate solution, shortening the optimization time.

The optimization algorithm stops in one of the following situations:

i) All *UD*s become zero for one candidate solution. This candidate solution is considered a global Pareto optimal point and it is the final solution.

ii) None of the candidate solutions can be further improved, meaning that the set of local Pareto optimal points was obtained. As final optimal solution we chose the one with the minimum value of the mean of unfulfillment degrees.

iii) Maximum number of iterations is reached.

The fuzzy optimization with qualitative gradients (GGFO) is fast, because it implies a low computation volume, but it is restricted to monotone function and can found only a local solution. Contrary, the fuzzy optimization with quantitative gradients (LGFO) and multiple search paths has the advantage to avoid a local solution and find a set of Pareto optimal points. Also it can cover a wide spectrum of the performance functions to be optimized. Due to these advantages this method is comparable or even better than some of the non-fuzzy methods.

7. Implementations and Results

The fuzzy optimization algorithms (see Fig. 1) are implemented in the Matlab environment, as CAD tools, with user-friendly graphical interfaces as communication bridges between the user and computer.

FMODO (Fuzzy Miller OTA Design Optimization) tool uses GGFO as optimization engine and it can be used to design Miller operational transconductance amplifiers (MOTA). FADO (Fuzzy Analog Design Optimization) is based on LGFO with multiple parallel search paths. FADO can be used to design several analog modules.

Due to the lack of space, we will discuss here only the implementation and some detailed results for FADO design tool.

7.1 FADO

FADO benefits by a library of three analog modules that can be designed: a simple CMOS operational transconductance amplifier (SOTA), a Miller compensated CMOS operational transconductance amplifier (MOTA) and a BJT common emitter stage (CE). The user can select the circuit to be designed. After selection, a new graphical interface, specific to the selected circuit opens.

7.1.1. MOTA Design Optimization

The schematic for the Miller operational transconductance amplifier is presented in Fig. 10. The design parameters of the circuit are: the dimensions of the transistors, the bias current I_b and compensation capacitance C_C. Not all circuit parameters are independent, so after a mathematical analysis just four independent design parameters result: I_b, $(W/L)1$, $(W/L)6$ and C_C. The circuit functions taken into consideration were: voltage gain A_{vo}, unity gain bandwidth GBW, phase margin *PM,* and slew rate *SR*.

Fig. 11 presents the FADO graphical interface to design the MOTA circuit.

The user can select all four available requirements or just some of them by activating the radio buttons. For the selected ones, the numerical values should be provided and the requirement type (“<”, “=” or “>”) should be selected.

The default values of maximum number of iterations (200) and of candidate solution (5) can be altered by the user. Some extra options of the tool

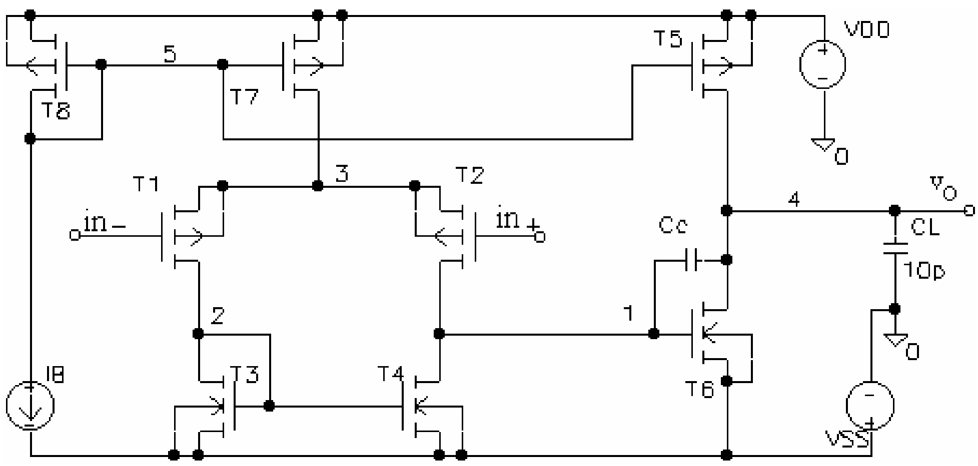

Fig. 10 Miller operational transconductance amplifier (MOTA)

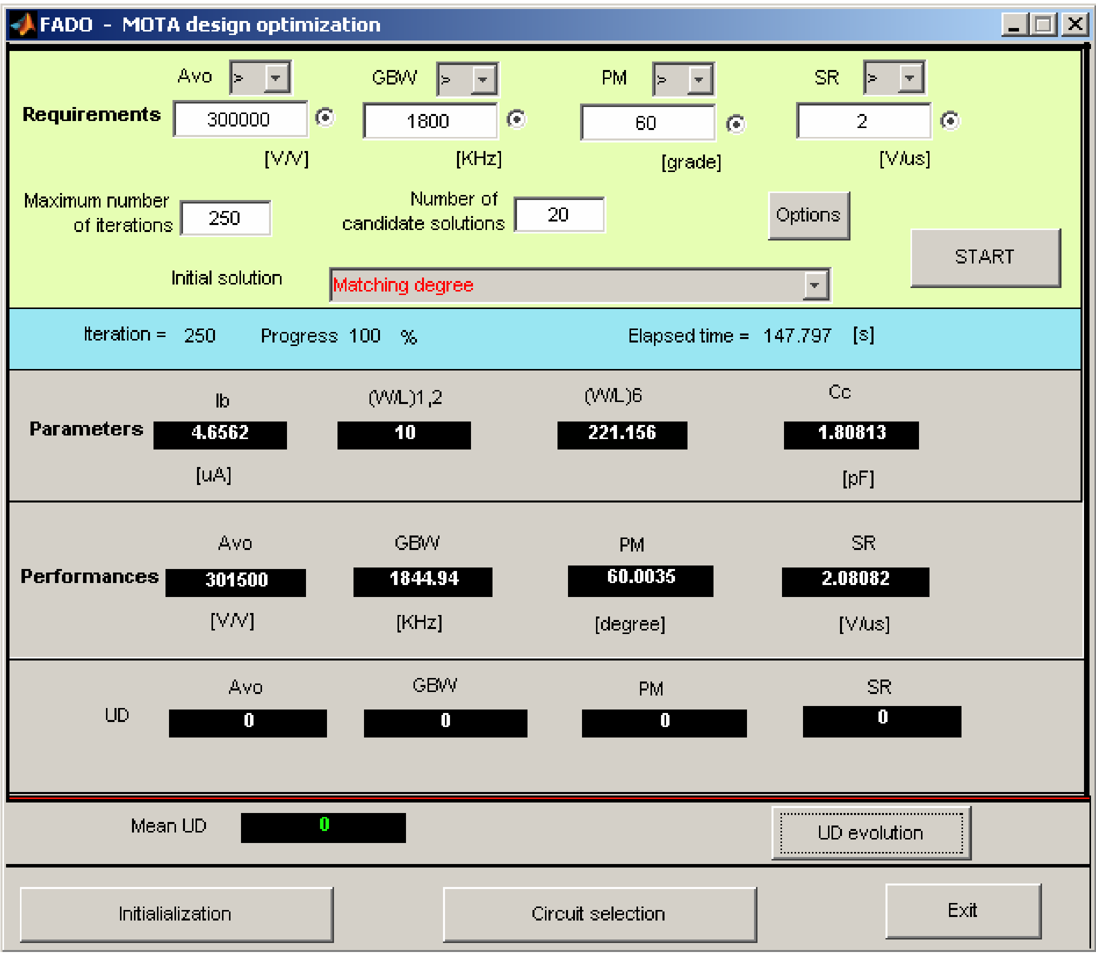

Fig.11 FADO graphical interface for design optimization of MOTA circuit

can be modified as well by pushing "Options" button.

The method to generate initial solutions (Matching degree, Latin Hypercube Sample, Random, User) has to be selected from the pop-up menu.

Optimization starts by acting the "Start" button. During the optimization, the number of current iteration, the progress ratio, and elapsed time are displayed.

To the end of the optimization, FADO provides the optimal solution (meaning the computed value of the parameter), the unfulfillment degree (*UD*) for each requirement and mean *UD*.

The time evolution of the mean *UD* and maximum

UD during the optimization are displayed if the user presses the “UD evolution” button.

The MOTA amplifier design is illustrated here for two sets of requirements: Set1, where all requirements are “greater than” type (Table 4.) and Set2 where one requirement is “equal” type (hard requirement), the other three requirements being “greater than” type (Table 6).

For Set1 of requirements, the optimization was run for a total number of 30 candidate solutions and a maximum number of 200 iterations. The results are presented in Table 4 (requirements, performances, mean *UD*s, iterations number) for three candidate solution (9, 12 and 26) .These candidate solutions provided very good results, especially candidate 12 that fulfills all the requirements (final mean *UD* = 0) after 104 iterations. The final solutions are different and they are part of the Pareto optimal set, specific to multiobjective optimization. Each of them can be selected by the user as final solution, according to the *UD* of the requirements and the value of the design parameters. Practically, all these three solutions accomplish the requirements alluding to real design usually accept some tolerances. It is obvious that the optimization strategy (LGFO with multiple search paths) has an appreciable chance to find the best optimal solution (global optimum), if such a solution exists.

Table. 4 Results for MOTA design optimization using FADO, Set1 of requirements

circuit function	require-ment	time moment	candidate solution 9		candidate solution 12		candidate solution 26	
			iterations: 29		iterations: 104		iterations: 24	
			perform	mean *UD*	perform	mean *UD*	perform	mean *UD*
A_{vo}	≥300000	initial	362421	initial 0.467	288359	initial 0.296	381886	initial 0.492
		final	**299326**		**301539**		**299697**	
GBW [kHz]	≥2000	initial	508.1		935.30		338.8	
		final	**1965.6**		**2000.08**		**1959.1**	
PM [°]	≥60	initial	81.4	**final 0.00071**	75.33	**final 0.0**	80.35	**final 0.0017**
		final	**58.6**		**61.36**		**57.07**	
SR [V/μs]	≥2	initial	0.55		1.09		0.38	
		final	**2.31**		**2.42**		**2.29**	

Table. 5 Comparative results for MOTA design optimization using FADO, fgoalattain, and FMODO

circuit function	require-ment	performances								
		candidate solution 9			candidate solution 12			candidate solution 26		
		FADO	fgoalattain	FMODO	FADO	fgoalattain	FMODO	FADO	fgoalattain	FMODO
A_{vo}	≥300000	299326	283762	293424	301539	296069	290654	299697	300852	292928
GBW [kHz]	≥2000	1965.6	1277	1953	2000.08	1820	1934	1959.1	1727	1950
PM [°]	≥60	58.6	64.15	58.57	61.36	60.73	57.99	57.07	60.09	58.47
SR [V/μs]	≥2	2.31	2.16	2.22	2.42	2.02	2.1	2.29	1.98	2.19
mean *UD*		0.00071	0.081	0.0009	0	0.005	0.0019	0.0017	0.012	0.0011
iterations		29	13	200	104	21	200	24	6	200

Table. 6 Results for MOTA design optimization using FADO, Set2 of requirements

circuit function	require-ment	time moment	candidate solution 18		candidate solution 25		candidate solution 45	
			iterations: 250		iterations: 250		iterations: 250	
			perform	mean *UD*	perform	mean *UD*	perform	mean *UD*
A_{vo}	=450000	initial	387906	initial 0.02	432221	initial 0.50	303209	initial 0.47
		final	**449235**		**449850**		**449338**	
GBW [kHz]	≥1500	initial	2092,7		502.2		544.6	
		final	**1540.0**		**1503.2**		**1495.6**	
PM [°]	≥60	initial	52.87	**final $2.2 \cdot 10^{-5}$**	78.90	**final $4.89 \cdot 10^{-6}$**	81.23	**final $5.7 \cdot 10^{-5}$**
		final	**59.66**		**59.93**		**60.45**	
SR [V/μs]	≥1,5	initial	2.30		0.54		0.58	
		final	**1.59**		**1.51**		**1.49**	

In its ordinary utilization, FADO automatically chooses as final solution the one having the minimum value of the mean *UD*, in our example the candidate solution 12 and the optimization stops at the iteration 104.
For the sake of comparison, Table 5 presents the performances and mean *UD* for MOTA design optimization obtained using our CAD tools (FADO and FMODO) and the "fgoalattain" optimization method from the Matlab Optimization Toolbox.
"Fgoalattain" is a multiobjective optimization method that implements the goal attainment method of Gembicki. The functions to be optimized by "fgoalattain" are our fuzzy models of circuit functions. Mean *UD* for the results provided by this method was subsequently computed using the fuzzy sets that define fuzzy objectives in FADO and FMODO. The initial solutions were the same for all three optimization methods.
Regarding the quality of final solutions, the superiority of our CAD tools compared with "fgoalattain" was confirmed (smallest *UD*). FADO and FMODO are comparable, an exception appearing for the candidate 12, where FADO proved to be superior by finding a solution with the realization of all design requirements.
From the convergence point of view, FADO and FMODO appear to be inferior compared with "fgoalattain", more iterations being necessary to rich their final solutions. The main advantage of FADO resides in its highly probability to find an as good as possible solution, due to the multiple search path.
For the Set2 of requirements, design optimization was carried out for a population of 30 candidate solutions, for a maximum number of 250 iterations. Table 6 presents the requirements, performances, mean *UD*s, to the beginning and to the end of the optimization. The results refer to a selection of three candidate solutions: 18, 25, and 45 that provided the best final solutions. FADO automatically selects as final optimal solution the one corresponding to the candidate 25 because of its lowest final mean *UD* ($4.89 \cdot 10^{-6}$). This solution assures full achievement for *GBW* (1503.2>1500) and *SR* (1.51>1.5) and almost full achievement for A_{vo} (449850 vis a vis >450000) and *PM* (59.93^o vis a vis >60^o).

7.1.2. SOTA Design Optimization

SOTA circuit (Fig. 6) was also considered for design optimization using FADO. Numerical values of the requirements and initial and final (after optimization) performances are presented in Table 7. All the requirements are considered as "greater or equal" type. FADO was set to run with a population of 30 candidate solutions. The optimal solution was found after only 9 iterations. All the performances fulfill the requirements, (the final mean *UD* is 0). We can see that the initial performances are pretty close to the requirements, so the optimization was very fast. This is a merit of the population of solutions, which enabled multiple points in the parameters space to search for the solution.

Table. 7 Results for SOTA optimization

circuit function	require-ment	time moment	perform	mean *UD*
Avo	≥50	initial	46.97	initial 0.271
		final	**52.20**	
GBW [kHz]	≥4500	initial	4245.9	
		final	**4585.2**	
PM [°]	≥60	initial	91.07	**final 0.0**
		final	**90.89**	
CMRR	≥1000000	initial	612111	
		final	**1000040**	

7.1.3. CE design optimization

The circuit of a simple CE stage is shown in Fig. 12. For this simple BJT amplifier we considered the following circuit functions: input resistance R_i, output resistance R_o, band-pass gain A_{vo}, and bandwidth *B*. For one resistor we choused a fixed value R_1=68 KΩ. The other resistors R_2, R_E, and R_C are the design parameters.

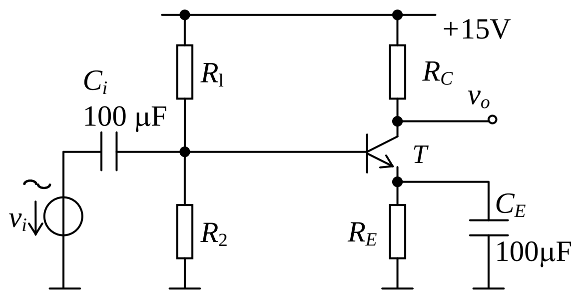

Fig. 12 CE amplifier

Table.8. Results for the CE optimization

circuit function	require-ments	time moment	perform	Mean *UD*
R_i [KΩ]	≥2.5	initial	1.32	initial 0.75
		final	**2.501**	
R_o [KΩ]	≤0.4	initial	0.86	
		final	**0.39**	
A_{vo}	≥30	initial	138.49	**final 0.0**
		final	**31.29**	
B[MHz]	≥150	initial	57.42	
		final	**158.64**	

The requirements for the R_i, A_{vo} and B are "greater or equal" type, while for the R_o is "less or equal" type. The results for CE design optimization using FADO are presented in Table 8. After 17 iterations, a solution that fully accomplishes all requirements was picked up from a number of 20 candidate solutions.

8. Conclusions

Two analog design optimization tools FMODO (Fuzzy Miller OTA Design Optimization) and FADO (Fuzzy Analog Design Optimization), embedding fuzzy techniques was introduced in this paper. Both tools benefit on some advantage offered by a collection of fuzzy approaches. The quality of initial solutions can be improved by selecting them according with their matching degrees in accomplishing the design requirements; these matching degrees being determined by means of fuzzy sets. Fuzzy sets used to define the optimization functions bring some advantages in comparison to classical methods: support real term, allows degrees of acceptability for solutions, and assure a known range [0, 1] for the value of objective functions. Takagi-Sugeno fuzzy systems are used to build fuzzy models of circuit performances. These fuzzy models are built automatically using numerical data sets. Main advantages of such fuzzy models are: high accuracy at a low computational cost and no restrictions for functions to be modeled. In the optimization engine, fuzzy systems are used to decide on the modification of every design parameter in each iteration. It worth to mention that these fuzzy systems incorporate human expert knowledge to guide the parameter modification toward optimal solutions during optimization. The combination of fuzzy objective functions and fuzzy optimization engines assures a real multiobjective optimization.

The results obtained with FADO, optimizing three analog modules are very promising. Optimal solutions are found in a reduced number of iterations, as for example 19, 104, or 24 iterations in the case of MOTA circuit design optimization ("≥" – type requirements). Very good results were obtained even if one requirement was "=" type, a more difficult design problem. In 250 iterations, FADO provides its optimal solutions having very low values of mean *UD* ($4.89 \cdot 10^{-6}$ or $2.2 \cdot 10^{-5}$). Due to the use of population of solutions we found a set of Pareto optimal points and the point with minimum *UD* has been choused as the final optimal solution. Also in the proximity of the final solutions the method works well to continue decrease *UD*s up to the local Pareto optimal point. The quality of each final solution is very high. This is possible because the method uses local gradient information and works in an adoptive manner: while the UD decrease, the step in the parameter modification also decreases.

References:

[1] S. Balkir, G. Dundar, G. Alpaydin, Evolution Based Synthesis of Analog Integrated Circuits and Systems, IEEE NASA/DoD Conference on Evolution Hardware, EH'04, ISBN 0-7695-2145-2/04, 2004. pp. 26-29.

[2] M. S. Baghini, R. Kanphade, D. G. Wakade, P. Gawande, M. Chhangani, M. Patil, GP-based Design and Optimization of a Floating Voltage Source for Low-Power and Highly Tunable OTA Applications, *WSEAS Transactions on Circuits And Systems,* Issue 10, Volume 6, ISSN: 1109-2734, October 2007, pp. 577- 582.

[3] R. A. Rutenbar, G. G. E. Gielen, J. Roychowdhury, Hierarchical Modeling, Optimization, and Synthesis for System-level Analog and RF Designs, *Proc. of the IEEE*, Vol. 95, Issue 3, March 2007, pp. 640-669.

[4] E. Shragowitz, J-Y.E. Lee, Q. Kang, Application of Fuzzy Logic in Computer-Aided VLSI Design, *IEEE Trans. on Fuzzy Systems*, vol.6. No 1, February 1998, pp.163-172.

[5] J. M. Mendel, Fuzzy Logic Systems for Engineering: A Tutorial, *Proc. of the IEEE,* Vol.83, No. 3, March 1995, pp. 345-377.

[6] I. Lagrat, A. El Ougli, I. Boumhidi, Optimal Adaptive Fuzzy Control for a Class of Unknown Nonlinear Systems, *WSEAS Transactions On Computer Research*, Issue 1, Vol. 3, ISSN: 1991-8755, Jan. 2008, pp. 89-98.

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

[8] F. Vallette, P. Garda, S. Feruglio, G. Vasilescu, Tolerances Analysis of MOSFET Integrated Circuits Performances, *WSEAS Transactions on Electronics*, Issue 12, Vol. 4, ISSN: 1109-9445, December 2007, pp. 277-283.

[9] B. Sahu, A. K. Dutta, Automatic Synthesis of CMOS Operational Amplifiers: A Fuzzy Optimization Approach, 15th Int. Conference on VLSI Design, 2002, pp. 366-371.

[10] M. Fares, B. Kaminska, FPAD: A Fuzzy Nonlinear Programming Approach to Analog Circuit Design, *IEEE Trans. on Computer-*

Aided Design of Integrated Circuits and Systems, vol. 14, no.7, July 1995, pp. 785-793.

[11] M. Hashizume, H.Y. Kawai, K. Nii, T. Tamesada, Design Automation System for Analog Circuits Based on Fuzzy Logic, IEEE Custom Integrated Circuits Conference, 1989, pp. 461-464.

[12] B. R. S. Rodriguez, M. A. Styblinski, Adaptive Hierarchical Multi-Objective Fuzzy Optimization for Circuit Design, IEEE International Symposium on Circuits and Systems, ISCAS '93, 1993, pp.1813-1816.

[13] L. Fu, J. Fan, L. Yuanchun, T. Yantao, D.Yisong, Study and Application of a Constrained Multi-objective Optimization Algorithm, IEEE International Vehicle Electronics Conference, IVEC 1999, pp. 305-307.

[14] M. Sakawa, Interactive Multiobjective Decision Making using Fuzzy Satisficing Intervals and its Application to an Human Water Resources System, North-Holland, Large Scale System, 10, 1986, pp. 203-213.

[15] G. Oltean, C. Miron, S. Zahan, M. Gordan, A Fuzzy Optimization Method for CMOS Operational Amplifier Design, Proc. of 5th Seminar on Neural Network, Neurel2000, Belgrade, 26-27 Sept 2000, pp. 152-157.

[16] G. Oltean, S. Hintea, Doris, Lupea, A Fuzzy Optimization Engine for Analog Circuit Design, Proc. of the 8th International Conference MIXDES2001, Zakopane, Poland, 21-23 June 2001, pp.77-82.

[17] G. Oltean, C. Miron, Emilia, Mocean, Multiobjective Optimization Method for Analog Circuits Design based on Fuzzy Logic, 9th IEEE International Conference on Electronics, Circuits and Systems - ICECS2002, Dubrovnik, Croatia, 15 – 18 Sept. 2002, pp. 777-780.

[18] G. Oltean, FADO - A CAD Tool for Analog Modules, Proc. of the International Conference on "Computer as a Tool" EUROCON2005, Belgrade, ISBN 1-4244-0050-3, IEEE catalog number: 05EX1255C, 2005, pp. 515-518.

[19] H. Y. Koh, C. H. Sequin, P. R. Gray, OPASYN: A Compiler for CMOS Operational Amplifiers, *IEEE Transactions on Computer-Aided Design*, vol.9, no.2, February 1990, pp. 113-125.

[20] A. Torralba, J. Chavez, , L. G. Franquelo, FASY: A Fuzzy-Logic Based Tool for Analog Syhesis, *IEEE Trans. Computer Aided Design*, vol.15, no7, July 1996, pp. 705-715.

[21] B. J. Sheu, A. H. Fung, Y-H. Lai, A Knowledge-Based Approach to Analog IC Design, *IEEE Transactions on Circuits and Systems*, vol. 35, no.2, February 1988, pp. 256-258.

[22] C. Michael, H. Su, M. Ismail, A. Kankunnen, M. Valtonen, Statistical Techniques for the Computer-Aided Optimization of Analog Integrated Circuit, *IEEE Transactions on Circuits and Systems-1; Fundamental Theory and Applications*, vol. 43, no. 5, May 1996, pp. 410-413.

[23] M. A. Aguirre, J. Chavez, A. Torralba, L. G. Franquelo, Analog Design Optimization by means of Tabu Search Approach, Proc of the IEEE Int. Symp. of Circuits and Systems ISCAS'94, London May 1994, Vol 1, pp. 375-378.

[24] R. Phelps, M. Krasnicki, R. A. Rutenbar, R. L. Carley, J. R. Hellums, Anaconda: Simulation-Based Synthesis of Analog Circuits via Stochastic Pattern Search., *IEEE Tran. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 19, No.6., June 2000, pp. 703-717.

[25] G. Oltean, Selection of the initial solutions based on matching degree in the frame of optimizing the analog design, Technical report, Technical University of Cluj-Napoca, RTH 2K001, 2001;

[26] T. Kiely, G. Gielen, Performance Modeling of Analog Integrated Circuits Using Least-Squares Support Vector Machines, Proc. of Design, Automation and Test in Europe Conference and Exhibition, Vol. 1, 2004, pp. 448-453.

[27] D. M. Binkley, Tradeoffs and Optimization in Analog CMOS Design, 14th International Conference on Mixed Design of Integrated Circuits and Systems, MIXDES07, 21-23 June 2007, pp. 47 – 60.

[28] R. S. Zebulum, M. A. Pacheco, M. M. Velasco, A Multiobjective Optimization Applied to the Synthesis of CMOS Operational Amplifier, Proc. of the Intl. Conf. on Microelectronics and Packaging, Brazil, Aug 1998.

[29] M. del Mar Hershenson, CMOS analog circuit design via geometric programming, American Control Conference, Volume 4, Issue , 30 June-2 July, 2004, pp. 3266 – 3271.

[30] M. del Mar Hershenson, S. P. Boyd, T. H. Lee, Optimal Design of a CMOS Op-amp via Geometric Programming, IEEE Trans. on Computer-Aided Design of Integrated Circuits and System, Vol. 20, No.1, January 2001, pp. 1-21.

[31] M. Barros, J. Guilherme, N. Horta, GA-SVM Optimization Kernel Applied to Analog IC Design Automation, Proc. of 13th IEEE International Conference on Electronics, Circuits and Systems, ICECS06, 2006, pp. 486-489.

[32] M. Barros, G. Neves, J. Guilherme, N. Horta, An evolutionary optimization kernel with adaptive parameters applied to analog circuit design, International Symposium on Signals, Circuits and Systems, ISSCS 2005, Volume 2, 14-15 July 2005, pp. 545 – 548.

[33] S. Zhao, L. Jiao, J. Zhao, Y. Wang, Evolutionary design of analog circuits with a uniform-design based multi-objective adaptive genetic algorithm, Conference on Evolvable Hardware 2005 NASA/DoD, 29 June-1 July 2005, pp. 26 – 29.

[34] K. Prakobwaitayakit, Analog circuit optimizer based on computational intelligence techniques, 47th Midwest Symposium on Circuits and Systems, MWSCAS, Volume 3, 25-28 July 2004, pp. iii - 291- 4.

[35] G. Oltean, L. Miclea, Mihaela, Gordan, Fuzzy Optimization Algorithm for Emitter Follower Design, Proccedings of the International Conference on Automation and Control, A&Q, Cluj-Napoca, May 28-29, 1998, pp. A535-A540.

[36] X. Ren, T. Kazmierski, Performance Modeling and Optimization of RF Circuits Using Support Vector Machines, Proc. of 14th International Conference on Mixed Design of Integrated Circuits and Systems, MIXDES07, 2007, pp. 317-321.

[37] F. De Bernardis, M. I. Jordan, A. Sangiovanni Vincentelly, Support Vector Machines for Analog Circuit Performance Representation, DAC 2003, Anaheim, California, USA, 1-58113-688-9/03/0006, 2003, pp. 964-969.

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

[39] N. D. Venkata, B. L Evans, An automated Framework for Multicriteria Optimization of Analog Filter Design, IEEE Tran. on Circuits and System-II: Analog and Digital Signal Processing, Vol. 46, No. 8, August 1999, pp. 981-990.

[40] G. Oltean, C. Miron, M. Crasi, Mihaela Carlugea, Evaluation of the Analog Circuits Performances Using Fuzzy Models, *Scientific Bulletin of the Politechnica University of Timisoara, Trans.on Electronics and Communications*, Vol 47(61) No 1, ISSN 1583-3380, 2002, pp. 12-17.

[41] K. Swings, W. Sansen, ARIADNE: A Constraint-Based Approach to Computer-Aided Synthesis and Modeling of Analog Integrated Circuits, *Analog Integrated Circuits and Signal Processing*, Kluwer Academic Publishers, Boston, 1993, pp. 37-55.

[42] P. C. Maulik, L. R. Carley, D. J. Allstot, Sizing of Cell-Level Analog Circuits Using Constrained Optimization Techniques, *IEEE Journal of Solid State Circuits*, vol. 28, no. 3, March 1993, pp. 233-240.

[43] A. Somani, P. P. Chakrabarti, A. Patra, An Evolutionary Algorithm-Based Approach to Automated Design of Analog and RF Circuits Using Adaptive Normalized Cost Functions, *IEEE Transactions on Evolutionary Computation*, Vol. 11, no. 3, 2007, pp. 336-353.

[44] T. Eeckelaert, R. Schoofs, G. Gielen, M. Steyaert, W. Sansen, Hierarchical Bottom-up Analog Optimization Methodology Validated by a Delta-Sigma A/D Converter Design for the 802.11a/b/g standard, Design Automation Conference, 43rd ACM/IEEE, 2006, pp. 25-30.

[45] G. Oltean, Multiobjective Fuzzy Optimization Method, *Scientific Bulletin of the Politechnica University of Timisoara, Trans.on Electronics and Communications*, Vol 49(63) No 1, ISSN 1583-3380, 2004, pag. 220-225.