Scientia Iranica, Vol. 9, No. 4, pp 419–424

SCIENTIA
IRANICA

Research Note

A Hybrid Reasoning-Based Tool for Analog Circuit Synthesis

F. Razaghian[1], S. Sadughi* and K. Badie[2]

In this paper, an intelligent CAD tool for the design of analog circuits is presented. First, the system uses fuzzy rule based reasoning to select alternative architectures for the circuits at each level of hierarchy and, then, deduces the device size using case-based reasoning, whose objective is to determine the plausible building blocks subspecifications on the basis of past design experience. In cases where some of the specifications have not been met, a qualitative reasoning-based approach is used to make final corrections. The above approach is demonstrated through the example of a CMOS Op-Amp.

INTRODUCTION

Design of analog integrated circuits is usually a time consuming process, due to the complex relations between design parameters and circuit performance [1]. The best approach for designing in the shortest possible time is to develop CAD-oriented tools that can automatically design analog circuits.

Analog circuit design is accomplished in two steps: In the first step, the designer selects an appropriate circuit topology from various possible topologies or architectures, in order to achieve a higher performance for a particular application. The second step is based on assigning values to circuit parameters so that circuit performance can satisfy a set of specifications. In this paper, a new approach is proposed for designing analog integrated circuits, based on the hybridation of three kinds of reasoning: fuzzy rule-based, case-based and qualitative.

Although the topology or architecture selection depends ordinarily on the designer's experience, it can, however, be automated with the aid of artificial intelligence methods [2,3]. It has been demonstrated that fuzzy logic, despite its intrinsic inexactitude, can act as a precise method for optimal decisions, including design problem solving [4]. The method proposed in this paper uses fuzzy rule-based reasoning to select an appropriate architecture out of the alternatives accumulated in a library. This guaranties the best possible initial guess and reduces the number of epochs for circuit simulation [5,6].

Case-based reasoning originally emerged as human memory and reminding [7]. This approach has been used to translate the input specifications into some building block subspecifications and subsequently to determine circuit device size efficiently.

The designed circuit is then simulated to show how efficiently it performs. If the conclusion is reached that some input specifications have not yet been met, a qualitative reasoning based approach [8-10], which is responsible for applying the most suitable (plausible trade-offs) variations to both circuit topology and device size is, subsequently, used to make the final correction on the circuit topology and the related device size. The design methodology is described in detail in the next section. Then, design samples as well as detailed results, are discussed.

AN OVERVIEW OF THE PROPOSED FRAMEWORK

The general structure of the framework is shown in Figure 1. It is comprised of three sections: Architecture selection, circuit synthesiser and circuit correction [11]. The proposed design system starts from input specifications and deduces the set of circuit parameter values.

In the first step, the program uses a fuzzy rule-based system to select the most appropriate architecture out of the alternatives existing in the library. The

1. *Department of Electrical Engineering, Islamic Azad University, Science and Research Branch, Tehran, I.R. Iran.*
*. *Corresponding Author, Department of Electrical Engineering, Sharif University of Technology, Tehran, I.R. Iran.*
2. *Iran Telecommunication Research Center, Tehran, I.R. Iran.*

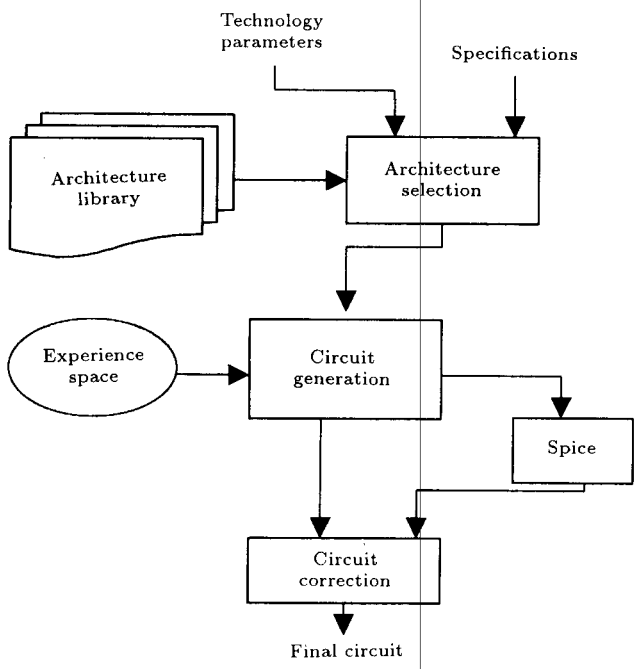


Figure 1. The general structure of the framework.

second step is to identify a first-cut initial solution in a hierarchical manner by using case-based reasoning. This procedure incorporates expert design knowledge and heuristics into a space that is named experience space. In this way, the architecture is defined in terms of a set of building-blocks. A preliminary evaluation of the first-cut solution is performed using a SPICE simulator, in order to decide whether or not it should be corrected. The circuit correction process is based on heuristics that stem from qualitative relationships between circuit performance and design variables.

This basic framework has many similarities with some knowledge-based design ideas. However, the major difference is that this framework characterizes the design strategy when faced with a known circuit architecture and a set of performance requirements by using case-based reasoning and experience space. Also, it is believed that a suitable mechanism for selecting among topological variants can be critical. Moreover, a tool capable of finding suitable sizes for a topology is still usable by analog designers.

Architecture Selection

Designing an analog circuit begins with design of the topology and approximate modelization of its behavior. Expert designers perform this task based on their experience. Therefore, in most cases, the designer selects one suitable architecture and then tries to modify it. The distinct feature of decision making based on fuzzy logic is its similarity with human reasoning and decision making. Fuzzy inference is based on a set of decision rules whose antecedents and consequents are linguistic terms. For instance, a rule can be defined as "If x is A_i and y is B_i, Then Z is C_i". The architecture selection is comprised of the following four principal components, as shown in Figure 2:

1. The fuzzification interface, by using triangular membership functions for the antecedents and singletons for the consequents, converts input data into a suitable linguistic value;
2. The knowledge-base contains necessary definitions, which are used to define linguistic terms such as "Large, Small,..." and a rule-base that characterizes the goals and control policy by means of a set of fuzzy rules that are combined by using the sentence connectives "and" and "also";
3. The decision-making logic has the capability of simulating human decision making, employing fuzzy implication and rules. A fuzzy rule is implemented by fuzzy implication R_i which is defined as follows:

$$\mu_{R_i} = \mu_{(A_i \text{ and } B_i \rightarrow C_i)} = (\mu_{A_i} \text{ and } \mu_{B_i}) \rightarrow \mu_{C_i}.$$

   Since each fuzzy rule is represented by a fuzzy relation, the overall behavior of the fuzzy system is characterized by these fuzzy relations. In other words, a fuzzy system can be characterized by a single fuzzy relation, which is the combination of the fuzzy relations in the rule set. The combination in question involves the sentence connective "also". Symbolically, $R = \text{also } (R_1, R_2, ..., R_n)$. In short, unlike the antecedent block, the result of the consequent block is a combination of all of the results of the rules. The technical term for determining this result is called "the maximum" because the maximum accepts all of the conclusions. In the proposed approach, the connective "and" and "also" are implemented by means of "minimum" and "maximum"[6];
4. Defuzzification is a mapping from a space of fuzzy actions onto a space of crisp actions. No systematic procedure exists for choosing the appropriate strategy. In the present approach, the Center Of Area (COA) strategy is chosen [4,5].

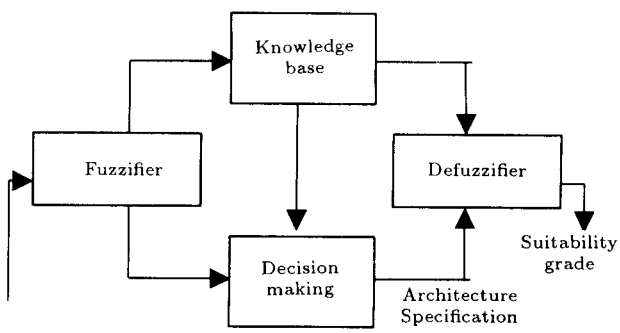


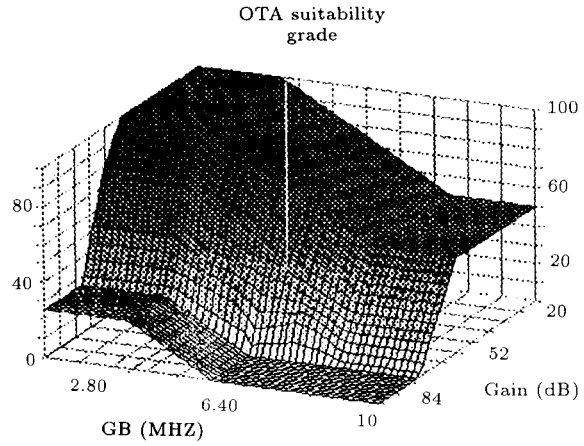
Figure 2. Architecture selection process.

The knowledge-base can be obtained from expert designers. The antecedent of the rules are needed performances such as, voltage gain, gain-bandwidth product slew rate, etc, while the consequent of the rules is the relative suitability of each architecture in the library. In Table 1, a set of rules given by an expert designer is presented for two architectures of op-amp (basic two stage and operational transconductance amplifier). By using the rules and inference process, the decision surfaces of Figure 3 are obtained. These figures show the suitability grade of each architecture with respect to some specification values.

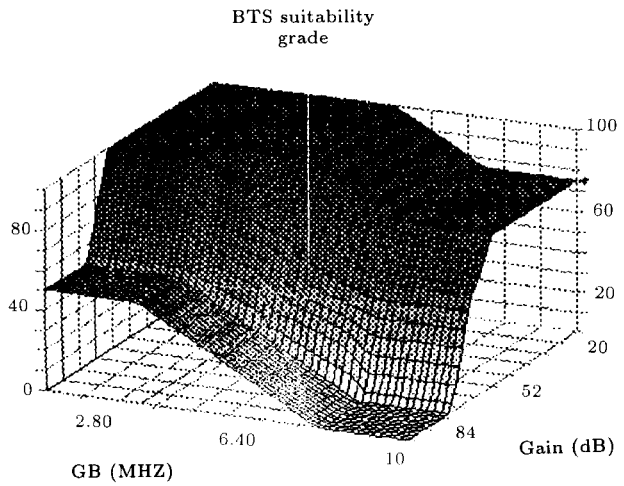
Table 1. Decision rules.

Gain	Small		Medium		Large	
GB	BTS	OTA	BTS	OTA	BTS	OTA
Small	VL	VL	VL	L	M	S
Medium	VL	L	L	M	S	VS
Large	L	M	M	S	VS	VS

VL: very large, L: large, M: medium, S: small, VS: very small



a) OTA



b) BTS

Figure 3. Decision surfaces.

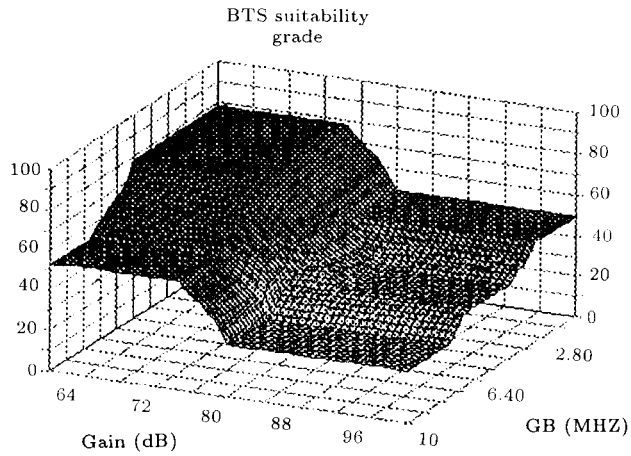


Figure 4. BTS decision surface obtained from automatic process.

As the fuzzy partitioning of the fuzzy input/output space is not deterministic and has no unique solution, a heuristic cut and trial procedure, via several simulations of different topologies, is performed to find the optimal fuzzy partition.

Using the above rules and the inference process, the suitability grade of each architecture at each level of hierarchy is calculated, with respect to its specifications and the architecture with the highest grade is selected as the most plausible one.

The system takes into account dc gain, gain bandwidth product, load capacitance, active area and the slew rate as well. As the number of specifications in the architecture selection process increases, decision rules are acquired with more difficulty, due to the existence of conflicting specifications. In these cases, the system is able to generate the decision rules automatically.

In this way, the specification space is divided into a number of cells, each one representing a set of specifications. For each cell and each possible architecture, the following cost function is optimized:

$$f(c) = \frac{1}{n}\sum_{1}^{n} \exp(\pm \frac{w_i(\text{spec}_i - \text{perf}_i)}{\text{spec}_i}).$$

The optimized value of the function is considered as the grade of the related architecture suitability.

Using this method, the decision surface of Figure 4 has been obtained for BTS architecture with 243 rules. The general behavior demonstrated in Figures 3b and 4 is similar, indicating that the experienced designer has a good feel for architecture selection.

Circuit Synthesis

In this section, the selected architecture is divided into smaller building blocks in a hierarchical manner. In the process of circuit architecture division, the required

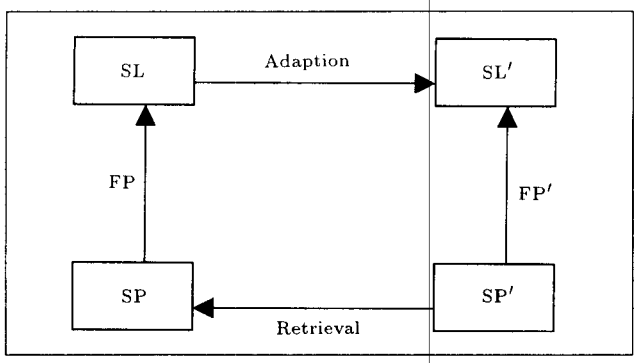


Figure 5. Transformation process in CBR. SP: Specification; SL: Solution; FP: First principles.

specifications should also be divided into some subspecifications for the smaller building blocks. In all existing systems, this translation is equation-based. But, even for a simple analog circuit, the exact equations are very complex and must be entered as thousands of lines of program code [12]. Moreover, adding the equations to a new circuit architecture requires a user who is a programmer, an analog designer and an expert intimate with the internal structure of the tool.

One of the central advantages in using a case-based approach for developing a knowledge-based system is that it can be applied to problems where a strong domain theory hardly exists [13]. In particular, CBR systems can be developed without explicit encoding of problem solving knowledge.

The basic tenet of CBR is that, rather than solving a problem from first principles or equations, it may be easier to retrieve a similar problem and transform the solution to that problem. Figure 5 shows this trade-off graphically.

Here, SP′ represents the specification for a new problem and SL' is the solution to that problem. FP′ represents the first principle reasoning that provides this solution. A CBR solution is worthwhile if the retrieval task, R, and the adaption task are both simpler and faster than FP′. So the major issue is the complexity of the adaptation task. The proposed framework uses an analogy function to retreive a similar past case via an experience space and an optimization algorithm for the adaptation task. In this manner, several different designs are performed and the solutions with the design paths are saved in the experience space. Since the stored paths are based on the expertise of experienced designers, they can produce results highly optimized in design efficiency.

Circuit Correction

Once the process of designing the circuit is simulated, the performance specifications will be compared with the simulated ones. In cases where the comparison requires some adjustment, the circuit is passed to a correction mode. The numerical optimizers, as tools for driving the designed circuit to an optimum point, are not suitable due to the following reasons:

1. The numerical optimizers, though blind, correct the circuit performance through excessive simulations;
2. They are not capable of correcting the circuit topology.

The qualitative reasoning, however, offers means which are out of reach for a numerical optimizer that can determine the effect of a circuit component's variations on circuit performance.

This approach uses knowledge which is specific to the failed performances, in order to indicate which parts of the architecture are responsible for the failure [6,8] and the gradient method is used to estimate the extent of adjustment.

The above reasoning ensures that solutions are obtained with less iterations because the stored knowledge provides useful information on the way the design performance can be corrected. On the other hand, if the qualitative dependencies are quantized into finer levels, such as "large increase", "small increase", etc., better correction is then expected to be achieved and the number of iterations may also become less, while the very ambiguities arising in the design process can be handled in a more convenient and efficient manner. The above approach was adopted in the proposed system.

DESIGN EXAMPLE

The desired specifications for an operational amplifier are shown in Table 2.

In this example, the gain and gain bandwidth specifications are considered as fuzzy objectives to select the best architecture. In this way, the selected architecture with a higher grade is shown in Figure 6.

Table 2. Op-amp input spec. and simulation results.

Performance	Specified	Predicted	Simulated
Voltage gain (dB)	80	86.89	88.5
Gain bandwidth product (MHz)	3	3.03	2.5
Phase margin (degree)	70	25.35	36
Slew rate (V/μ s)	3	3.06	3.03
Pos. out. swing (V)	4	3.79	4.8
Neg. out swing (V)	-4	-4.21	-4.8
CMR (+)	+3	3.4	3.8
CMR (-)	-3	-2.52	-2.46
Power dissipation (mW)	10	1.41	1.39
Area (μm^2)	10000	3870	
Architecture selection grade	BTS=49.5 OTA=25.14		

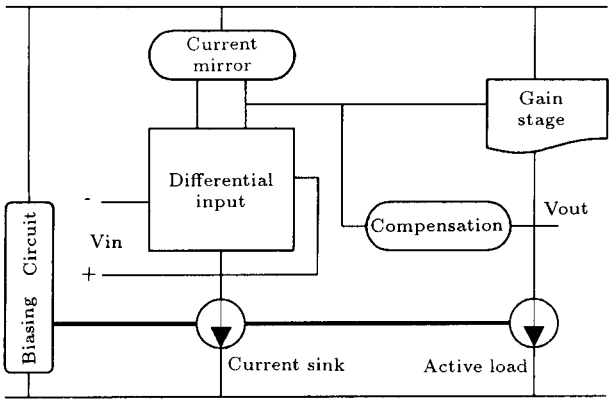


Figure 6. The selected architecture.

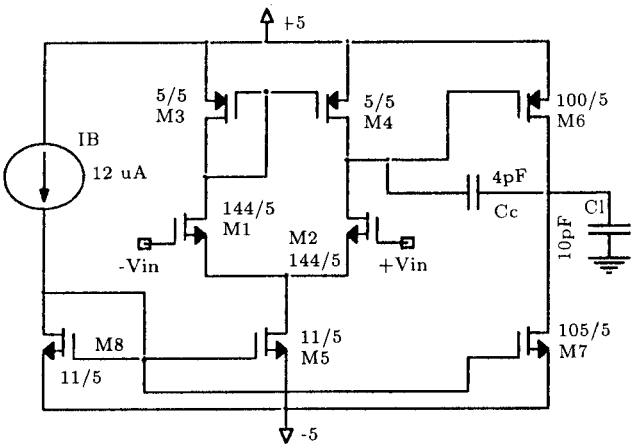


Figure 7. The synthesized architecture.

The circuit is designed in a hierarchical manner. At each step, the lower-level architectures are selected using fuzzy rules and device sizing is done via a case-based approach. The synthesized circuit is shown in Figure 7. Table 2 also illustrates a comparison between the results of applying the proposed approach and the simulated results obtained through using SPICE. As seen, the phase margin and the gain bandwidth requirements are not satisfied. The results from this simulation were processed by a correction module and the system found that the necessary transistor sizes for correcting the phase margin quantitatively would have been unacceptably large. Therefore, in the first iteration the system attempted to implement the correction with the gain bandwidth product. The results are shown in Table 3a.

These new results are processed again for correcting the phase margin. In the second iteration, the compensation network was modified by adding a nulling resistor. The corrected circuit was simulated and its specifications were now met. The final circuit is shown in Figure 8 and its performance is listed in Table 3b. The designed circuit was then stored as a case in the experience space.

Table 3a. Results after first correction.

Performance	Predicted	Simulated
Gain (dB)	89.78	88.14
GB (MHZ)	3.62	4.34
PM (deg)	51.76	55.5
SR (V/μ s)	3.06	4.05
Vout (max)	3.96	4.8
Vout (min)	-4.21	-4.8
CMR (+)	3.967	3.9
CMR (-)	-2.87	-3.9
Power diss	2.52	2.46

Table 3b. Results after second correction

Performance	Predicted	Simulated
Gain (dB)	89.78	91.5
GB (MHZ)	3.62	3.6
PM (deg)	90.72	91
SR (V/μ s)	3.06	3.07
Vout (max)	4	4.8
Vout (min)	-4.2	-4.8
CMR (+)	3.56	3.9
CMR (-)	-2.87	-3.8
Power diss	2.52	2.46

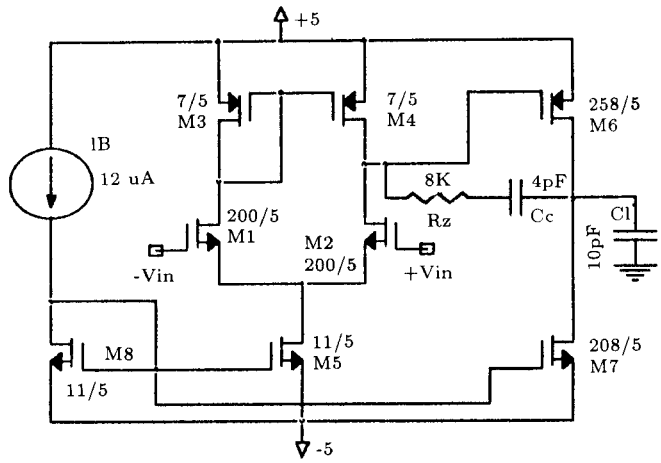


Figure 8. The final circuit.

Here, let a new set of specifications be assumed that are similar to that in Table 1, except for voltage gain (120dB). As the most similar case, the system quickly selected the circuit from experience space shown in Figure 8. For achieving the desired voltage gain, the circuit was processed by correction rules. The new circuit is shown in Figure 9 and its performance is listed in Table 4.

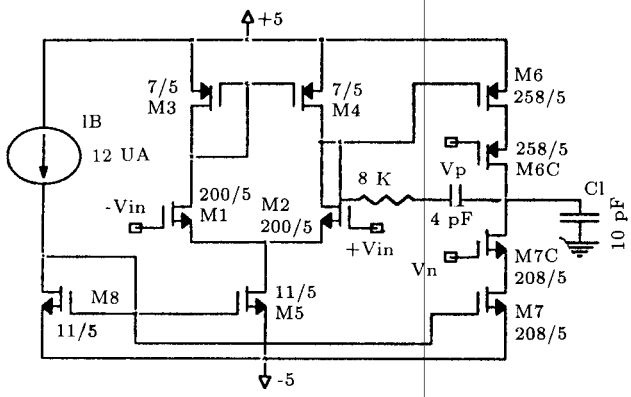


Figure 9. The fast circuit designed by using CBR.

Table 4. The performance values of circuit in Figure 9.

Performance	Predicted	Simulated
Gain (dB)	131.14	129
GB (MHZ)	3.62	3.55
PM (deg)	90.72	88.75
SR (V/μ s)	3.06	3.09
Vout (max)	2.87	4.45
Vout (min)	-3.46	-4.9
CMR (+)	3.56	3.9
CMR (-)	-2.87	-3.8
Power diss	2.52	2.39

CONCLUSION

In this paper, an approach was proposed for the design of analog circuits using hybrid reasoning. By using a fuzzy rule-base reasoning, the design problem was formulated in a way that reflects the human approach to architecture selection in the design process. The decision rules set can be introduced by an expert designer or may be obtained by a learning procedure, which uses the past design experiences. An effective methodology was also proposed for analog circuit design that stores and reuses previous designs by using case-based reasoning via an experience space.

Experimental results show a significant improvement in design efficiency when CBR is used. These experiments also show that a designer can improve the experience space more broadly through it's iterative utilization, which boosts design productivity. Obviously, the method is efficient only if sufficient cases from previous designs are available.

Also, it is indicated that qualitative reasoning-based correction can provide a suitable base for improving design performance without any need for iterative simulation. It can also assist the numerical optimizer in the fine tuning of the design process. The proposed system, based on the above approach, was used to design unbuffered CMOS op-amps. The results support the ability of the system for a fast design process, which include necessary modifications. The proposed approach can equally be applied to design of other analog circuits such as comparators, voltage references, etc.

REFERENCES

1. Allen, P.E. and Holberg, D.R., *CMOS Analog Circuit Design*, HRW (1987).
2. Koh, J.Y. and Gray, P.R. "OPASYN: A compiler for CMOS operational amplifiers", *IEEE Trans. CAD*, **9**(2), pp 113-123 (1990).
3. Bowman, R.J. and Lane, D.J. "A knowledge-based system for analog circuit design", *Proc. IEEE Int. Conf. Computer-Aided Design*, pp 210-212 (1985).
4. Ross, T.J., *Fuzzy Logic with Engineering Applications*, McGraw-Hill (1997).
5. Torralba, A., Chavez, J. and Franquelo, L.G. "FASY: A fuzzy-logic based tool for analog synthesis", *IEEE Trans. CAD*, **15**(7), pp 705-715 (1996).
6. Sadughi, S., Razaghian, F. and Badie, K. "Automated design of analog integrated circuits based on fuzzy logic and artificial intelligence", *Proceeding of World Automation Conference (WAC)*, Hawaii, USA (2000).
7. Kolodner, J.L., *Case-Based Reasoning, Mountainview CA*, Morgan Kaufmann, pp 369-382 (1993).
8. Toumazou, C. and Makris, C.A. "Analog IC design automation: Part 1,2", *IEEE Trans. CAD*, **14**(2), pp 218-254 (1995).
9. Razaghian, F. and Sadughi, S. "Intelligent design of integrated circuits based on qualitative reasoning", *Proc. of the 4th Annual International CSI Computer Conference (CSICC)*, pp 213-218 (1998).
10. Harjani, R., Rutenbar, R.A. and Carley, L.R. "OASYS: A framework for analog circuit synthesis", *IEEE Trans. CAD*, **8**(12), pp 1247-1265 (1989).
11. Razaghian, F., Sadughi, S. and Badie, K. "An intelligent framework for designing analog circuits based on hybrid reasoning", *Proceeding of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp 189-192 (2000).
12. Degrauwe, M.G.R. et al. "Towards an analog system design environment", *IEEE J. Solid-State Circuits*, **24**(3), pp 659-671 (1989).
13. Kolodner, J.L. "What is case-based reasoning", *Proc. on CBR in the 8th National Conference on AI*, pp 1-32 (1990).