FUZZY COMPUTING OF INITIAL SOLUTION IN ANALOG CIRCUIT DESIGN OPTIMIZATION

Gabriel OLTEAN
Technical University of Cluj-Napoca, Str. C. Daicoviciu No.15, Cluj Napoca, Romania
Gabriel.Oltean@bel.utcluj.ro

Abstract: A new fuzzy based method to compute the initial parameter values for the optimization design phase of electronic circuits is presented. A Mamdani type fuzzy system models the relations expressing the dependencies between circuit parameters and circuit functions. This fuzzy system is able to compute the values of design parameters for every set of design requirements. This way, the optimization phase has a good, automatic computed, starting point. Using this approach for a CMOS operational amplifier, initial solutions were computed for several sets of design requirements. Some of our results are much better or comparable than the ones presented in the literature.

Keywords: fuzzy system, model, CAD, electronic circuit, requirement, performances, parameters.
FUZZY COMPUTING OF INITIAL SOLUTION IN ANALOG CIRCUIT DESIGN OPTIMIZATION

Gabriel OLTEAN
Technical University of Cluj-Napoca, Str. C. Daicoviciu No.15, Cluj Napoca, Romania
Gabriel.Oltean@bel.utcluj.ro

Abstract: A new fuzzy based method to compute the initial parameter values for the optimization design phase of electronic circuits is presented. A Mamdani type fuzzy system models the relations expressing the dependencies between circuit parameters and circuit functions. This fuzzy system is able to compute the values of design parameters for every set of design requirements. This way, the optimization phase has a good, automatic computed, starting point. Using this approach for a CMOS operational amplifier, initial solutions were computed for several sets of design requirements. Some of our results are much better or comparable than the ones presented in the literature.

Keywords: fuzzy system, model, CAD, electronic circuit, requirement, performances, parameters.

I. INTRODUCTION

Computer Aided Design (CAD) system has become an important tool for designing mixed analog/digital integrated circuits. However, the design of analog part of such a system, which is usually small compared to the digital part, is still a bottom-neck task in the entire design due to the complex relation between circuit performance and design parameters.

The analog design process consists of several phases, for example: topology selection, parameter optimization, layout generation, extraction, etc [1].

In the analog circuit design process there is a specific terminology:
- **parameters** – the quantities whose values are to be determined (e.g. length and width of MOS transistors, bias current, compensation capacitor);
- **circuit functions** – e.g. dc gain, gain bandwidth, slew rate, etc.;
- **performances** – values of the circuit functions computed by analysis/simulation;
- **requirements** - values of the circuit functions imposed by design specifications.

The parametric optimization is a technique used to select the circuit design parameter values in such a way that the actual circuit meets the design requirements. Regardless of optimization methods (classical optimization methods, knowledge-based methods, global optimization methods, convex optimization methods [2]), a special attention should be paid to the initial solution of the design parameters.

Without a good starting point, an optimization run may converge very slowly or converge to a local minimum whose performance is significantly worse than the circuit’s best capability [3].

In the optimization algorithms presented in the literature there are some techniques for obtaining the initial solution. In most proposed techniques this critical task is left to the user [4], [5], [6]. The user should be an experienced designer, not a novice one, in order to choose an appropriate solution. Other technique use approximate analytic relations that model the circuit behavior. Then, design equations (parameters depending on requirements) are extracted and used, together with basic assumption about circuit, to compute the initial parameter values [7], [5], [8]. In some others techniques, the initial solution can be randomly chosen [9], [3].

In this paper a fuzzy logic based method to determine an initial solution is proposed. For every set of design requirements a fuzzy system will select a set of parameter values conducting to circuit performances as close as possible to the requirements.

II. FUZZY COMPUTING OF INITIAL SOLUTION

In the circuit design optimization a very important problem is the choice of the starting point. The quality of the optimized solution and the design time depend heavily on a good starting point. Generally speaking, every electronic circuit can be analyzed and usually nonlinear approximate expression for circuit function can be found. In the design step we need design equations that express design parameters depending on requirements. Finding these expressions is not such an easy task. Moreover, usually, the number of design parameters and the number of design equations are quite different, making the solution more complicated.

In order to overcome these drawbacks we proposed a fuzzy logic based approach that model the design parameters-requirements relations. (Fig. 1.) A Mamdani type fuzzy logic system computes a set of appropriate
values for design parameters for every set of requirements. The fuzzy logic system is built based on a set of numerical data points obtained by simulation.

<table>
<thead>
<tr>
<th>design requirements</th>
<th>design parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy Logic System</td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 1. Parameters – requirements model**

Everybody know that an electronic circuit can be very accurate simulated using a software simulator (SPICE). In simulation, for a set of parameter values we can obtain the values of the circuit functions. This way we have a data point with parameters as inputs and with performances as outputs. All we have to do is to take such data points (but considering performances as inputs and parameters as outputs) and use them to build our fuzzy system.

Our fuzzy system is build through the following steps:

1. The reference electronic circuits' simulator SPICE is used to generate a set of input (parameters)–output (performance) data points. To obtain the sets of the parameter values for simulation we use the Latin Hypercube SampleTechnique [10].

2. The data points are subdued to a subtractive clustering algorithm. The clustering algorithm returns the position of cluster centers and the sigma value that specifies the range of influence of cluster centers in each of the data dimensions.

3. The Mamdani type fuzzy system is build considering each cluster as a fuzzy rule with circuit performances in the antecedent part and with circuit parameters in the consequence part. The number of rules is equal to the number of cluster. For each variable the fuzzy sets are gaussian curves with the median value (corresponding to a membership degree \( \mu = 1 \)) equal to the corresponding cluster center. The sigma value is the same as that specifying the range of influence of the center cluster.

For example, let us consider a very simple circuit with two design parameters \( x_1 \) and \( x_2 \) and two circuit functions \( f_1(x_1,x_2) \) and \( f_2(x_1,x_2) \) (shortly \( f_1 \) and \( f_2 \)). Let’s suppose further that after simulation and subtractive clustering we obtained two clusters with centers: \((x_{11}, x_{21}, f_{11}, f_{21})\), respectively \((x_{12}, x_{22}, f_{12}, f_{22})\). For \( x_1 \) and \( f_1 \) the fuzzy sets can be as in Fig. 2. In the same manner may result the fuzzy sets for \( x_2 \) and \( f_2 \).

The fuzzy rule base has the following two rules:

R1: IF \( f_1 \) is \( mf1 \) AND \( f_2 \) is \( mf1 \) THEN \( x_1 \) is \( mf1 \), \( x_2 \) is \( mf1 \)

R2: IF \( f_1 \) is \( mf2 \) AND \( f_2 \) is \( mf2 \) THEN \( x_1 \) is \( mf2 \), \( x_2 \) is \( mf2 \)

The values of the design parameters should result in such a way that all the design requirements are satisfied as closely as possible. So, because the antecedent part of each rule contains all the input variables, I believe the activation degree of the rule should be the arithmetic mean of partial activation degree. As a result I used the \( \text{mean} \) operator for AND method. Also I used \( \text{prod} \) operator for the implication method and \( \text{max} \) operator for the aggregation method.

As in most fuzzy system, choosing the right defuzzification method is a very important thing. In our system there can be situations in which the activation degrees of the two most important fuzzy rules are quite similar. These two rules can have different fuzzy sets in their consequence for a certain parameter. (Fig. 3.) This situation can really happen because there can be multiple solutions (design parameters) for the same requirements. In above-mentioned case the two rules represent two different solutions. We only have to choose one of them.

Using centroid defuzzification we find the value \( x_{1c} \), that is not a good one, representing an average between two solution. By way of consequence we choose the \( \text{mom} \) (mean of maxima) defuzzification method. This way we find the value \( x_{1m} \) which is the right one, representing the most appropriate solution.

**III. RESULTS**

In order to validate the proposed fuzzy computing of initial solution, we present in this section an example for a basic two stage Miller compensated operational amplifier shown in Fig. 4.

As independent design parameters we consider \((W/L)1=r1\) for \( T1 \), \((W/L)6=r6\), for \( T6 \), \( Ib \) and \( Cc \). We consider fixed values for two parameters \((W/L)5 = 10\) for \( T5 \) and \((W/L)7 = 1\), for \( T7 \). The others design parameters result from the following relations:
Fig. 3. Defuzzification

Fig. 4. Basic two stage CMOS operational amplifier
The expression for \( (W/L)_3 = (W/L)_4 \) results from

\[
(W/L)_3 = (W/L)_4 = \frac{1}{2} \frac{(W/L)_5 \cdot (W/L)_7}{(W/L)_6} = (W/L)_4
\]

because the \( V_{GS} \) voltage for \( T_5 \) and \( T_7 \) is the same and \( I_5 = I_7 = I_4 / 2 \). In order to provide a maximum symmetry in the input stage (same voltage in the nodes 1 and 2) we should have \( V_{GS1} = V_{GS2} = V_{GS5} \) [11]. Also we use \( V_{DD}=2.5V \), \( V_{SS}=-2.5V \) and \( CL=10pF \).

We consider the following circuit functions: DC gain (avo), Gain Bandwidth (gbw), Phase Margin (pm) and Slew Rate (sr).

To generate the fuzzy system that model the requirements-parameters relations we followed the modeling procedure described in section II.

1. We obtained 550 data points using SPICE simulation: 550 AC analysis for avo, gbw and pm and 550 transient analysis for sr.
2. For the clustering, we used the subtractive clustering algorithm implemented in the Fuzzy Toolbox of Matlab [12]. First, I set the parameters of subtractive clustering algorithm to obtain 11 clusters.
3. The fuzzy logic system has 11 fuzzy rules and was implemented using the Fuzzy Toolbox of Matlab too. The structure of the resulting fuzzy system is presenting in Fig. 5.

In Table 1.1 are illustrated the results (performances and relative errors) obtained for five sets (1\(^{\text{st}} \) to 5\(^{\text{th}} \)) of input design requirements. With the specified values of requirements we obtain the parameters (initial solution for design optimization) presented in Table 1.2. The performances in Table 1.1 are those obtained by SPICE simulation using the parameters in Table 1.2. Also in Table 1.1. we compute the modulus of relative error of every performance compared with the corresponding requirement. For every set of requirements we compute an mean error (last row in Table 1.1.) as the arithmetic mean of the modulus of the relative errors.

The relative errors are quite large for some set or requirements (for example 68.05% for set 5\(^{\text{th}} \), 47.75% for set 3\(^{\text{rd}} \) and 40.79% for set 1\(^{\text{st}} \)). In order to reduce these errors we tried to increase the number of rules in our fuzzy system, increasing the number of clusters in the subtractive clustering. The results obtained using 21 fuzzy rules are presented in Table 2.1. and Table 2.2. for the same sets of requirements as in Table 1.1.

To lighten the comparison we represented on the same diagram (Fig. 6) the mean errors for the five sets of requirements for both fuzzy system. The solution obtained is better in the case of 21 fuzzy rules than for 11 fuzzy rules for most of the requirement sets, especially for 1\(^{\text{st}} \) (13.98% compared with 40.79%) and 5\(^{\text{th}} \) (60.72% compared with 68.05%). For the 2\(^{\text{nd}} \) set the mean error is slightly larger for 21 rules (13.2%) compared with the one obtained for 11 rules (11.09%).

The proposed fuzzy method acts, up to a point, in a similar manner as a look-up table. Moreover, if the set of input values (requirements) is not in the table the fuzzy system is able to choose a response from the table that fit as good as possible to the requirements, as in the cases of the previous presented results. If the requirement

---

**Fig. 5** The structure of the fuzzy logic system

**Fig. 6** Mean relative error
### Table 1.1. Requirements, performances and relative errors for 11 fuzzy rules

<table>
<thead>
<tr>
<th>Circuit functions</th>
<th>1(^0)</th>
<th>2(^0)</th>
<th>3(^0)</th>
<th>4(^0)</th>
<th>5(^0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avο [V/µA]</td>
<td>432000</td>
<td>377452</td>
<td>12.62</td>
<td>505413</td>
<td>378002</td>
</tr>
<tr>
<td>GbW [KHz]</td>
<td>1250</td>
<td>421</td>
<td>66.32</td>
<td>720</td>
<td>633</td>
</tr>
<tr>
<td>Pm [°]</td>
<td>60</td>
<td>64.54</td>
<td>7.57</td>
<td>78</td>
<td>81.2</td>
</tr>
<tr>
<td>Sr [V/µs]</td>
<td>1.5</td>
<td>0.35</td>
<td>76.67</td>
<td>0.67</td>
<td>0.65</td>
</tr>
<tr>
<td>Error [%]</td>
<td>40.79</td>
<td>11.09</td>
<td>47.75</td>
<td>25.06</td>
<td>68.05</td>
</tr>
</tbody>
</table>

### Table 1.2. Initial solution for 11 fuzzy rules

<table>
<thead>
<tr>
<th>Parameters</th>
<th>1(^0)</th>
<th>2(^0)</th>
<th>3(^0)</th>
<th>4(^0)</th>
<th>5(^0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ib [µA]</td>
<td>0.728</td>
<td>4.18</td>
<td>3.03</td>
<td>4.23</td>
<td>3.03</td>
</tr>
<tr>
<td>r1</td>
<td>2.19</td>
<td>7.46</td>
<td>4.44</td>
<td>8.44</td>
<td>4.44</td>
</tr>
<tr>
<td>r6</td>
<td>298</td>
<td>404</td>
<td>255</td>
<td>72.8</td>
<td>255</td>
</tr>
<tr>
<td>Cc [pF]</td>
<td>1.57</td>
<td>5.39</td>
<td>0.592</td>
<td>0.396</td>
<td>0.592</td>
</tr>
</tbody>
</table>

### Table 2.1. Requirements, performances and relative errors for 21 fuzzy rules

<table>
<thead>
<tr>
<th>Circuit functions</th>
<th>1(^0)</th>
<th>2(^0)</th>
<th>3(^0)</th>
<th>4(^0)</th>
<th>5(^0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avο [V/µA]</td>
<td>377452</td>
<td>432000</td>
<td>12.62</td>
<td>378002</td>
<td>505413</td>
</tr>
<tr>
<td>GbW [KHz]</td>
<td>421</td>
<td>1250</td>
<td>66.32</td>
<td>633</td>
<td>720</td>
</tr>
<tr>
<td>Pm [°]</td>
<td>78</td>
<td>60</td>
<td>7.57</td>
<td>81.2</td>
<td>7.57</td>
</tr>
<tr>
<td>Sr [V/µA]</td>
<td>0.67</td>
<td>1.5</td>
<td>76.67</td>
<td>0.65</td>
<td>1.5</td>
</tr>
<tr>
<td>Error [%]</td>
<td>13.98</td>
<td>13.2</td>
<td>46.78</td>
<td>24.93</td>
<td>60.72</td>
</tr>
</tbody>
</table>

### Table 2.2. Initial solution for 21 fuzzy rules

<table>
<thead>
<tr>
<th>Parameters</th>
<th>1(^0)</th>
<th>2(^0)</th>
<th>3(^0)</th>
<th>4(^0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ib [µA]</td>
<td>1.74</td>
<td>2.7</td>
<td>1.4</td>
<td>4.23</td>
</tr>
<tr>
<td>r1 (w/l)1</td>
<td>3.36</td>
<td>6.97</td>
<td>8.63</td>
<td>8.44</td>
</tr>
<tr>
<td>r6 (w/l)6</td>
<td>298</td>
<td>375</td>
<td>174</td>
<td>72.8</td>
</tr>
<tr>
<td>Cc [pF]</td>
<td>0.984</td>
<td>4.41</td>
<td>3.24</td>
<td>0.396</td>
</tr>
</tbody>
</table>
coincide with the coordinates of a cluster center the results are very good: for two such selected cases the mean relative error was 0.5% and 0.24% respectively.

For comparison, in Table 3., the result obtained by Fares and Kaminska [5] for the same circuit, using approximate analytic design equation to find the initial solution, is presented.

Table 3. Result from [5]

<table>
<thead>
<tr>
<th>functions</th>
<th>requirements</th>
<th>perform</th>
<th>error [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>avo</td>
<td>10000</td>
<td>16351</td>
<td>63.51</td>
</tr>
<tr>
<td>gbw [KHz]</td>
<td>3000</td>
<td>2800</td>
<td>6.67</td>
</tr>
<tr>
<td>pm [°]</td>
<td>45</td>
<td>48.5</td>
<td>7.78</td>
</tr>
<tr>
<td>sr [V/µs]</td>
<td>6</td>
<td>5.63</td>
<td>6.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>21.03</td>
</tr>
</tbody>
</table>

We can see that our results are much better (1°, 2°), comparable (4°), or worse (3°, 5°) than the one presented in [5]. However, the comparison can be only roughly made because the design parameters are different, the requirements sets are not the same, etc.

IV. CONCLUSION

We have shown how fuzzy logic can be used to compute the initial values of the design parameters for the optimization phase of an electronic circuits. This way, an unexperienced designer can use this method to obtain a good initial starting point, in an automatic manner. To validate the proposed method, we used it to find the initial solution for design optimization of a CMOS operational amplifier. The result shows that our method can provide much better, or comparable solution than the one obtained in [5] using approximate design equations.

As a future task, we will use the initial solution generated with this fuzzy method as a starting point in optimizing the design of a CMOS operational amplifier. We are very confident in obtaining very good results: a short time for optimization (i.e. a small number of iterations) and a good final parameter values which determine circuit performances very close to the design requirements.

ACKNOWLEDGMENTS

The author wish to thanks Prof. Costin Miron and ass. prof. Mihaela Gordan for their valuable suggestions and comments that helped him to continue and finalize this work.

REFERENCES