A Formal Approach to Verification of Linear Analog Circuits with Parameter Tolerances

Lars Hedrich Erich Barke
Institute of Microelectronic Systems
University of Hanover, Germany
hedrich@ims.uni-hannover.de

Abstract

This contribution presents an approach to formal verification of linear analog circuits with parameter tolerances. The method proves that an actual circuit fulfills a specification in a given frequency interval for all parameter variations. It is based on a curvature driven bound computation for value sets using interval arithmetic. Some examples demonstrate the feasibility of our approach.

I. Introduction

Formal verification, in particular equivalence checking is an attractive alternative to simulation and very popular in the digital world today. Its main concept - to prove that two circuits have the same in-/output behavior - can also be transferred to analog circuits. In this paper we will present an approach to formal verification of linear analog circuits. Basically, two circuit descriptions on different levels of abstraction with parameter tolerances are compared. The approach verifies the following hypothesis:

> For all frequencies in a given frequency interval and for all parameter variations within the given parameter tolerances the first or actual (act.) circuit fulfills the specification described by the second or specifying (spec.) circuit having also parameters given in tolerances.

Many other approaches have been proposed to compute performance characteristics from an actual circuit and to compare it with given specifications, like *yield estimation* [1], *worst-case analysis* [2, 3, 4] or *design centering* [5]. Most of these approaches are based on probabilistic methods giving good and reliable results for the mentioned tasks. However, they are not able to prove, that a circuit with tolerance parameter fulfills a certain specification for all parameter combinations, because the tolerance parameters are modelled with a probabilistic density function, or the computed regions are non pessimistic approximations like convex polyhedrons. In contrast to the above, the proposed approach deals with intervals, enabling an exact proof of the correctness of the design.

The main idea of our procedure is to compare the value sets of the complex transfer functions H_{spec} and H_{act}. Related work in computing value sets can be found in [7, 8, 9,]. All these methods restrict the transfer functions to special classes, for example linear or multilinear dependent transfer functions [8], while our approach deals with a much larger class of transfer functions. Additionally, most of them deal only with outer bound computations of the value sets.

II. Circuit Description

Linear analog circuits can be described by transfer functions. Symbolic analysis methods [6] are able to calculate transfer functions from netlists resulting in a parameterized description of circuit behavior

$$\mathrm{H}(s,\mathbf{p}), \quad \mathbf{p} = [p_1, p_2, \ldots, p_n], \tag{1}$$

where $\mathbf{p}$ is the parameter vector of n parameters. Here, each parameter tolerance is given as an interval defining the parameter interval vector

$$\boldsymbol{p} = \left\{ p_i = [\underline{p}_i .. \overline{p}_i], i = 1..n \right\}. \tag{2}$$

Interval variables are printed in italics and interval vectors are printed in bold italics. The parameters are assumed to be independent and belong to a finite real interval. $\boldsymbol{p}$ is an n-dimensional hypercube in the parameter space.

We compute a safe approximation of the value sets $A = \left\{ \mathrm{H}(s = j\omega_0, \mathbf{p}) \,\middle|\, \mathbf{p} \in \boldsymbol{p} \right\}$ for each given transfer function. A safe approximation for the actual transfer function at a fixed frequency is an outer bound of the value set

$$\tilde{A}_{\mathrm{act,safe}} \supseteq A_{\mathrm{act}} = \left\{ \mathrm{H}_{\mathrm{act}}(\mathrm{s} = \mathrm{j}\omega_0, \mathbf{p}_{\mathrm{act}}) \middle| \, \mathbf{p}_{\mathrm{act}} \in \boldsymbol{p}_{\mathrm{act}} \right\}. \quad (3)$$

A safe approximation at a fixed frequency $\mathrm{s} = \mathrm{j}\omega_0$ for the specifying transfer function is an inner bound of the value set

$$\tilde{A}_{\mathrm{spec,safe}} \subseteq A_{\mathrm{spec}} = \left\{ \mathrm{H}_{\mathrm{spec}}(\mathrm{s} = \mathrm{j}\omega_0, \mathbf{p}_{\mathrm{spec}}) \middle| \, \mathbf{p}_{\mathrm{spec}} \in \boldsymbol{p}_{\mathrm{spec}} \right\}. (4)$$

In Figure 1 an example for the outer and inner bound $\tilde{A}_{\mathrm{act,safe}}$ and $\tilde{A}_{\mathrm{spec,safe}}$ is shown.

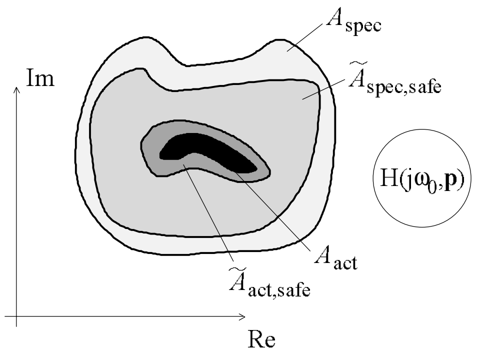

A_{spec}
Im
$\tilde{A}_{\mathrm{spec,safe}}$
$\mathrm{H}(\mathrm{j}\omega_0,\mathbf{p})$
A_{act}
$\tilde{A}_{\mathrm{act,safe}}$
Re

Figure 1: Value sets A_{spec}, A_{act} and bounds $\tilde{A}_{\mathrm{act,safe}}$, $\tilde{A}_{\mathrm{spec,safe}}$ for $\mathrm{H}_{\mathrm{spec}}$, $\mathrm{H}_{\mathrm{act}}$ at a fixed frequency

If $\tilde{A}_{\mathrm{act,safe}} \subset \tilde{A}_{\mathrm{spec,safe}}$ holds, $\mathrm{H}_{\mathrm{act}}$ fulfills $\mathrm{H}_{\mathrm{spec}}$ at a specific frequency ω_0. The extension to a proof for a frequency interval is described in Section V.

III. Outer Bound

An outer bound of a transfer function value set can be calculated by an interval extension using complex interval arithmetic [10, 11]:

$$\tilde{A}(\boldsymbol{p}) \supset A(\boldsymbol{p}) = \left\{ \mathrm{H}(\mathrm{s} = \mathrm{j}\omega_0, \mathbf{p}) \middle| \, \mathbf{p} \in \boldsymbol{p} \right\}. \quad (5)$$

Dividing the interval vector $\boldsymbol{pp}$ into a union of subinterval vectors $\boldsymbol{p}_{\mathrm{k}}$ according to $\boldsymbol{p} = \bigcup_{\mathrm{k}=1}^{\mathrm{m}} \boldsymbol{p}_{\mathrm{k}}$ the unified outer bound $\tilde{A}^{'}(\boldsymbol{p}) = \bigcup_{\mathrm{k}=1}^{\mathrm{m}} \tilde{A}_{\mathrm{k}}(\boldsymbol{p}_{\mathrm{k}})$ for $\mathrm{m} \to \infty$ converges to the tight or exact bound of the value set $A(\boldsymbol{p})$ (see [12]). Based on these results we compute an as tight as possible outer bound for the actual transfer function $\mathrm{H}_{\mathrm{act}}$ by subdividing the intervals recursively. For each subdivision the algorithm chooses a parameter that leads to the smallest overestimation. This technique automatically divides only those parameter intervals that contribute to the overestimation. If a parameter does not have any impact on the overestimation, it will not be divided leading to much less subdivided intervals and very small overestimation.

After subdivision, the resulting complex intervals are geometrically combined into a single circumscribing polygon (see Figure 2 for an example). This polygon describes an outer boundary $\partial\tilde{A}$ for the bound $\tilde{A}$ of the exact value set A. In most cases the proposed algorithms deal with boundary description by polygons. However, for the ease of understanding, the regions themselves are used instead of their boundaries in the remainder of the paper.

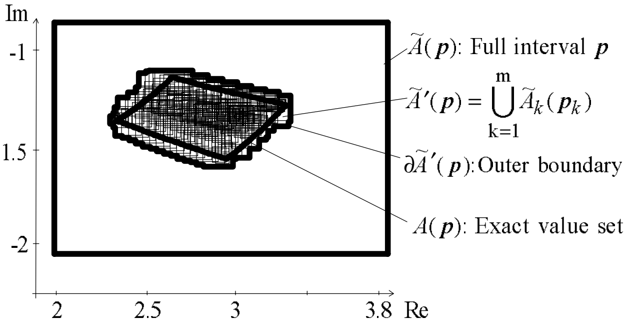


Figure 2: Computation of an outer bound of the function:

$$H = \frac{a}{(1+b^2)\cdot 0.1\cdot \mathrm{j} + 0.1\cdot a + 1}\;, \; a = [4.5..6],\; b = [-0.4..\,0.6]$$

For transfer functions with many parameters the problem of overestimation is still an substantial property of the algorithm. However, if e.g. the transfer function is given in linear dependent form (with the parameters as independent coefficients of the nominator and denominator polynoms, [7, 13]), a more efficient algorithm for calculating the outer bounds can be selected, for example the algorithm explained in Section IV.

IV. Inner Bound

From the viewpoint of interval analysis the calculation of an inner bound is a much bigger problem, because the inclusion property of the interval arithmetic [11] only facilitates the safe computation of outer bounds. However, if we restrict the class of functions to those having a simply connected region for the complex value set, an inscribing polygon can be constructed which is proven to belong to the value set.

A. Safe Path Between Two Points

The goal of this method is to find a region containing the actual parameterized curve between two endpoints.

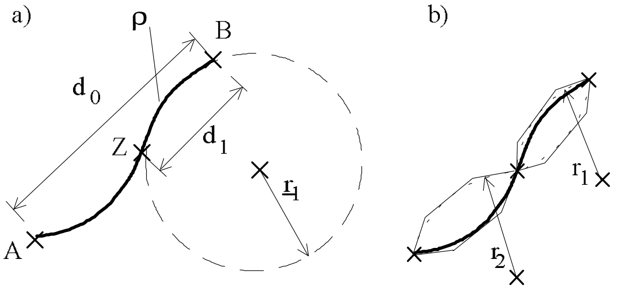


Figure 3: a: Maximum curvature and intermediate point Z on a curve ρ, b: safe path region approximated by a polygon

In Figure 3 a) an (unknown) curve ρ (plotted in bold) with its endpoints A and B is shown. The curve ρ is given by a complex function F(q) of one parameter $q = [\underline{q}..\overline{q}]$, while the endpoints can be calculated according to A={Re(F($\underline{q}$)), Im(F($\underline{q}$))}, B={Re(F($\overline{q}$)), Im(F($\overline{q}$))}. The radius of curvature of the parameterized curve ρ is defined by

$$r(q) = \frac{(R' + I')^{\frac{3}{2}}}{R' \cdot I'' - I' \cdot R''},\ R' = \frac{\partial \mathrm{Re}(F(q))}{\partial q},$$
$$R'' = \frac{\partial R'(q)}{\partial q}, I' = \frac{\partial \mathrm{Im}(F(q))}{\partial q}, I'' = \frac{\partial I'(q)}{\partial q}. \quad (6)$$

Using interval arithmetic a lower bound $\underline{r}$ for the radius of curvature can be computed. If it is smaller than half the distance between the two endpoints A and B ($\frac{1}{2} \cdot d_0 > \underline{r}_1$ in Figure 3 a)) the interval q is divided into two intervals by inserting an additional point (Z) at half the interval of q ($Z = \left\{ \mathrm{Re}(F(\frac{\underline{q}+\overline{q}}{2})), \mathrm{Im}(F(\frac{\underline{q}+\overline{q}}{2})) \right\}$). The generated intervals $[\underline{q}..\frac{\underline{q}+\overline{q}}{2}]$, $[\frac{\underline{q}+\overline{q}}{2}..\overline{q}]$ are then processed in the same way as the original interval. If the radius of curvature is larger than half the distance between the endpoints (i.e. $r_1 > \frac{1}{2} \cdot d_1$), the region of the actual parameterized curve ρ can be bounded by two segments of a circle which is pessimistically approximated by a polygon consisting of 3 lines (see Figure 3 b).

B. Inner Bound of a Value Set Using Curvature Examination

Assuming that the value set A of a complex function H is a simply connected region, every closed path, which lies safely within the value set, encloses a part of the value set. By means of the algorithm in Section IV A, such a path and the resulting inner bound of the value set can be computed as follows:

```
Inner Bound (F(p),p)
  Divide parameter hypercube into a set of 2-dimensional faces
  Map the faces in the H(jω)-space
  Compute the enveloping polygon E of the faces in the H(jω)-space:
    $E = [e_1, e_2, .. e_k]$
  for each segment $(e_i, e_{i+1})$ of the polygon E do
    Compute curvature driven path $((e_i, e_{i+1}), F(\boldsymbol{p}),$
        $\boldsymbol{p} = [p_{e_i} .. p_{e_{i+1}}])$
  return inscribing polygon of $\tilde{A}$
```

Figure 4: Algorithm for computing an inner bound

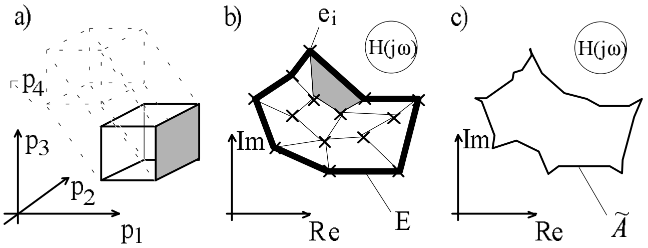


Figure 5: a: Faces of the 4-dimensional hypercube in the parameter space, b: faces and enveloping polygon mapped to the H(jω)-space, c: resulting inscribing polygon

In Figure 5 a simple example is presented in order to explain the main steps of the algorithm.

Because the enveloping polygon E is determined in a sequence with counterclockwise steps, the safe path can be computed using only an inner polygon approximation of the circle segment in order to get a pessimistic bound. The result is a polygon which describes the safe inner bound $\tilde{A}$ by its boundary.

One reason for computing the enveloping polygon from the image of the parameter cube faces and not from the corners using a simple convex hull approach is, that along the edges of the faces only one parameter is varying and all others are constant. This results in a much lower overestimation for the interval computation of the radius of curvature and, therefore, reduces computation time and complexity.

V. Extension to Proof within Frequency Interval

With the previously defined algorithms for computing inner and outer bounds, a proof that the actual circuit fulfills a specifying circuit is possible at a single frequency ω_0. In order to extend the proof to a given frequency interval, this frequency interval is introduced as an additional special interval parameter.

An approach handling the whole frequency interval in one step is not feasible, because a 3-dimensional calculation and a proof that the actual tube V_{act} lays inside the specifying tube V_{spec} would necessarily be a computationally tough problem. Hence, the frequency interval is divided into several small intervals and for each interval a projection of the frequency interval onto the H(jω) plane at one frequency is carried out resulting in a 2-dimensional pessimistic but simple comparison for each small frequency interval.

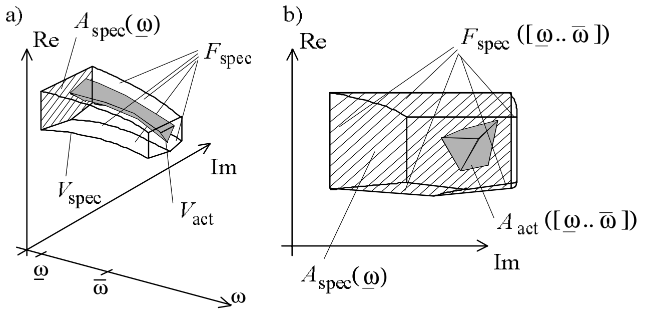

a)
$A_{\mathrm{spec}}(\underline{\omega})$
Re
F_{spec}
Im
V_{spec}
V_{act}
$\underline{\omega}$
$\overline{\omega}$
ω
b)
Re
$F_{\mathrm{spec}}\ ([\underline{\omega}..\overline{\omega}])$
$A_{\mathrm{act}}\ ([\underline{\omega}..\overline{\omega}])$
Im
$A_{\mathrm{spec}}(\underline{\omega})$

Figure 6: a: 3-dimensional frequency interval,
b: 2-dimensional projection

In Figure 6 a) the 3-dimensional value set for a frequency interval: $\omega = [\underline{\omega}..\overline{\omega}]$ is shown. Figure 6 b) displays the corresponding 2-dimensional projection. The dark shaded volume V_{act} is mapped to the dark shaded region A_{act}. The face $A_{\mathrm{spec}}(\underline{\omega})$ is hatched in Figure 6 a) and b). The other areas in Figure 6 are the faces of H_{spec} which extend into the frequency interval. They are called the frequency faces $F_{\mathrm{spec},p_{\mathrm{i}}}(\omega, p_{\mathrm{i}}, \mathbf{p})$ and have only one parameter p_{i} and ω in an interval. In addition to the proof that $A_{\mathrm{act}}(\omega)$ is inside $A_{\mathrm{spec}}(\underline{\omega})$, all frequency faces must not touch $A_{\mathrm{act}}(\omega)$.

In Figure 7 the formal verification algorithm is presented. Note, that it tries to calculate the outer bound for the faces using curvature examination (algorithm of Section IV B) because this method computes a tighter outer bound much faster than the direct interval method (algorithm of Section III). In order to use this algorithm for determining outer bounds, it has to be proved that the value set is bounded by its edges. This is checked for the faces $F_{\mathrm{spec},p_{\mathrm{i}}}(\omega, p_{\mathrm{i}}, \mathbf{p})$ having two interval parameters ω, p_{i} by investigating the perpendicular area vector being strictly positive or negative using interval arithmetic:

$$\mathrm{n} \subset \mathrm{R}^{+} \ \vee \ \mathrm{n} \subset \mathrm{R}^{-},$$

$$\mathrm{n} = \frac{\partial \mathrm{Re}(\mathrm{F}_{\mathrm{spec},\mathrm{p_i}})}{\partial \omega} \cdot \frac{\partial \mathrm{Im}(\mathrm{F}_{\mathrm{spec},\mathrm{p_i}})}{\partial \mathrm{p_i}} - \frac{\partial \mathrm{Re}(\mathrm{F}_{\mathrm{spec},\mathrm{p_i}})}{\partial \mathrm{p_i}} \cdot \frac{\partial \mathrm{Im}(\mathrm{F}_{\mathrm{spec},\mathrm{p_i}})}{\partial \omega}. \quad (7)$$

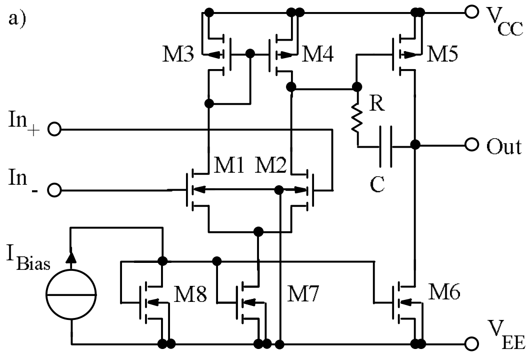
```
Formal Verification ($H_{spec}$, $H_{act}$, $\omega = [\underline{\omega}..\overline{\omega}]$ , $\boldsymbol{p}_{\mathbf{spec}}$, $\boldsymbol{p}_{\mathbf{act}}$)
  Divide $\omega$ in k parts [$\omega_1$, $\omega_2$,..., $\omega_k$]
  for each $\omega_i$ do
    Compute $\tilde{A}_{act,safe}$ = Outer Bound ($H_{act}$, $\omega_i$, $\boldsymbol{p}_{\mathbf{act}}$)
    Compute $\tilde{A}_{spec,safe,\underline{\omega}_i}$ = Inner Bound ($H_{spec}$, $\underline{\omega}_i$ , $\boldsymbol{p}_{\mathbf{spec}}$)
    Compute $\tilde{A}_{spec,safe,\overline{\omega}_i}$ = Inner Bound ($H_{spec}$, $\overline{\omega}_i$ , $\boldsymbol{p}_{\mathbf{spec}}$)
    for all edges $p_j$ of $\tilde{A}_{spec,safe,\underline{\omega}_i}$ do
      if $n \subset R^+$ or $n \subset R^-$ then
        Compute $\tilde{F}_{spec,safe,p_j}$ = Outer Bound using
           curvatures ($H_{spec}$, $\omega_i$, $p_{j,spec}$, $\mathbf{p_{spec}}$) (Section IV)
      else
        Compute $\tilde{F}_{spec,safe,p_j}$ = Outer Bound ($H_{spec}$, $\omega_i$,
           $p_{i,spec}$, $\mathbf{p_{spec}}$) (Section III)
    if not ( $\tilde{A}_{act,safe} \subseteq \tilde{A}_{spec,safe,\underline{\omega}_i}$ $\wedge$
      $\forall p_j$: $\tilde{A}_{act,safe} \cap \tilde{F}_{spec,safe,p_j} = \varnothing$ ) then
      if frequency interval is too small then
        return false
      else
        Reduce frequency interval $\omega_i$
  return true
```

Figure 7: Formal verification algorithm

VI Experimental Results

The algorithm is implemented using the symbolic math package Maple V™ [14]. The circuits are first linearized at their operating point and afterwards analyzed with a symbolic analyzer in order to get the transfer function including the parameters in symbolic form. Due to the prototype implementation and the use of Maple V™ the CPU times of the algorithm are still high. A more efficient implementation will reduce them drastically.

Two examples are presented in order to show the feasibility of the approach. The first example deals with a CMOS operational amplifier (actual circuit), which is verified against a behavioral specification at a high level of abstraction. The OP consists of 8 MOS transistors (see Figure 8). The MOS transistors are modelled by a simple nonlinear model including technology parameters. In this example, the W/L ratios of all single or paired transistors on the signal path constitute the tolerance (see Table 1). The behavioral model of the specification (spec. circuit) is given as a transfer function (see Figure 8). The parameters and their tolerances are presented in Table 1.



b)

$$H(s) = \frac{A_0 \cdot e^{j\cdot\Delta\varphi}}{1 + A_0\frac{\sin(phm)}{gbw\cdot 2\pi}\cdot s + \frac{A_0\cdot\cos(phm)-1}{gbw^2\cdot 4\pi^2}\cdot s^2}$$

Figure 8: a: Schematic of the OP versus b: behavioral description

Parameter	Nominal	Interval
W/L of M1, M2	8.8	8.712 .. 8.888
W/L of M3, M4	29.3	29.04 .. 29.62
W/L of M5	13.33	13.17 .. 13.46

Parameter	Symbol	Specific.	Interval
Gain	A_0	> 93 dB	50,000 .. 500,000
Gain-bandwidth	gbw	> 1 MHz	1 MHz .. 10 MHz
Phase margin	phm	< 60°	40° .. 60°
Phase tolerance	$\Delta\varphi$	< 5°	-5° .. +5°

Table 1: Parameters and their tolerances of the actual circuit and specifying behavioral model

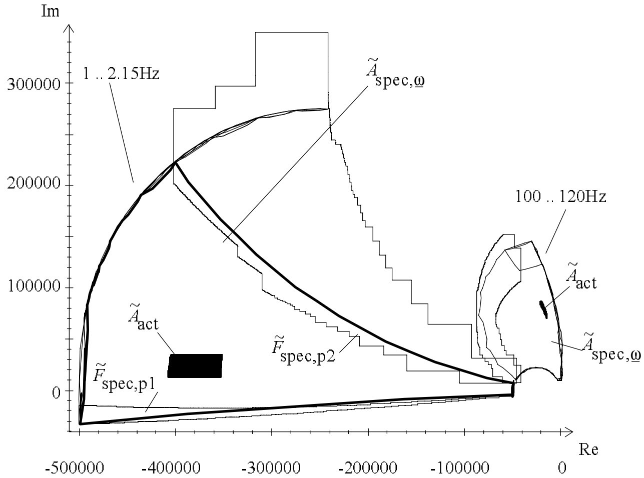
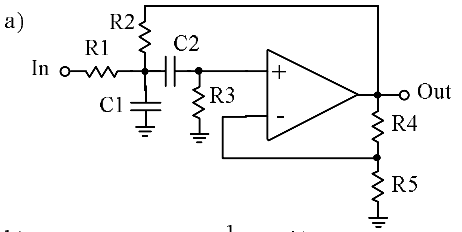


Figure 9: Computed bounds for the MOS-OP at two frequency intervals: 1 .. 2.15 Hz and 100 .. 120 Hz

The verification process starts with a desired frequency interval from 1 Hz to 6 MHz resulting finally in over 100 frequency intervals. Two intervals at the frequencies 1 .. 2.15 Hz and 100 .. 120 Hz are given in Figure 9. Some areas and bounds are identified as follows. $\tilde{A}_{act}$ is the outer bound of the actual circuit, $\tilde{A}_{spec,\underline{\omega}}$, is the inner bound of the specifying circuit at the lower frequency of the frequency interval, $\tilde{F}_{spec}$ are the outer frequency faces of certain parameters. In this case the algorithm terminates successfully. A change in circuit parameters will lead to a negative verification result. In this case the designer gets hints from the relative positions of the areas.

A second example deals with a bandpass. A netlist is verified versus a behavioral description of a bandpass. The netlist and the transfer function of the behavioral description are shown in Figure 10.



b)

$$H(s) = \frac{A_0\cdot k\cdot\frac{1}{\omega_g}\cdot s\cdot e^{j\Delta\varphi}}{1+\frac{1}{\omega_g}(3-k)\cdot s+\frac{1}{\omega_g^2}\cdot s^2}$$

Figure 10: a: Netlist and b: behavioral description of the bandpass example

The netlist has 5 resistors which are assumed to vary correlated with one relative parameter r_{tol} and to have an additional independent variance for each independent resistor($R2_{tol}$,$R3_{tol}$,$R4_{tol}$). The two capacitors are modelled in the same way. The tolerance parameters for the two circuits are:

Parameter	Interval
r_{tol}	1 ± 1%
$R2_{tol}$	1 ± 0.1%
$R3_{tol}$	1 ± 0.1%
$R4_{tol}$	1 ± 0.1%
c_{tol}	1 ± 1%
$C1_{tol}$	1 ± 0.1%

Parameter	Symbol	Specification	Interval
Gain	A_0	1 ± 5 %	0.95 .. 1.05
Center frequency	ω_g	1 kHz ± 5%	950 Hz .. 1.05 kHz
Phase tolerance	$\Delta\varphi$	< 10°	-10° .. +10°

Table 2: Parameters and their tolerances of the bandpass netlist and the specifying behavioral bandpass model

The algorithm starts with a frequency interval from 100 Hz to 10000 Hz. In Figure 11 the areas and bounds for 7 selected frequency intervals are shown. The entire frequency interval is splitted by the algorithm into 210 intervals. Note, that in this example and in the previous one the frequency faces $\widetilde{F}_{\text{spec}}$ are computed mostly using the curvature algorithm resulting in a much better approximation of the areas.

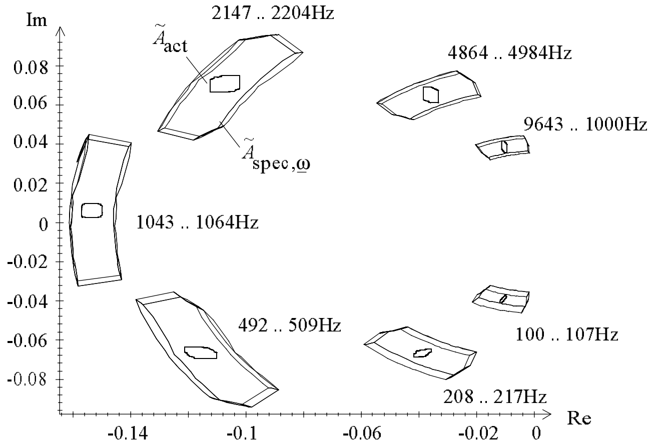


Figure 11: Results from the bandpass example at 7 different frequency intervals

VII Conclusion

In this contribution we have presented an approach to formal verification of linear analog circuits with parameters given in tolerances. The main goal is to give an automatic method for proving that an actual circuit fulfills a specification. Additionally, we focus on making as less restrictions to the class of handled circuits as possible. With respect to these goals the disadvantage of having exponential time complexity in terms of the parameters for the actual circuit has to be accepted. For the smaller class of linear dependent circuits (parameters as independent coefficients of the nominator and denominator polynoms) the curvature driven bound algorithm with linear time complexity can be chosen.

The proposed approach enables the user to verify his implementation against the specification without doing multiple simulations. In the failure case the value set in the complex H(jω) plane graphically supports the user in finding actual problems in the circuits. Finally, the algorithm can easily be extended to prove the stability of the circuits using the zero inclusion condition [15].

References

[1] St.W. Director, P. Feldmann, K. Krishna, "Optimization of Parametric Yield: A Tutorial", Proc. Custom Integrated Circuits Conference, pp. 3.1.1-3.1.8, 1992

[2] A. Dharchoudhury, S.M. Kang, "Worst-Case Analysis and Optimization of VLSI Circuit Performances", IEEE Trans. on Computer-Aided Design, vol. 14, no. 4, pp 481-192, 1993

[3] K.J. Antreich, H.E. Graeb, C.U. Wieser, "Circuit Analysis and Optimization Driven by Worst-Case Distances", IEEE Trans. on Computer-Aided Design, vol. 13, no.1, pp. 57-71, 1994

[4] G.E. Müller-L., "Limit Parameters: The General Solution of the Worst-Case Problem of the Linearized Case", IEEE Intern. Symp. on Circuits and Systems, vol. 3, pp. 2256-2259, 1990

[5] St.W. Director, G.D. Hachtel, "The Simplical Approximation Approach to Design Centering", IEEE Trans. on Circuits and Systems, vol. 24, no. 7, pp. 363-372, 1977

[6] G. Gielen, W. Sansen, "Symbolic Analysis for Automated Design of Analog Integrated Circuits", Kluwer Academic Publishers, Boston, 1991

[7] A. Levkovich, E. Zeheb, N. Cohen, "Frequency Response Envelopes of a Family of Uncertain Continuous-Time Systems", IEEE Trans. on Circuits and Systems, vol. 42, no. 3, pp. 156-16, May 1995

[8] S.P. Bhattacharyya, H. Chapellat, L.H. Keel, "Robust Control: The Parametric Approach", Prentice Hall, London, 1995

[9] P.-O. Gutman; C. Baril, L. Neumann, "An Algorithm for Computing Value Sets of Uncertain Transfer Functions in Factored Real Form ", IEEE Trans. on Automatic Control, vol. 39, no. 6, pp. 1269-73, 1994

[10] R.E. Moore, "Interval analysis", Prentice Hall, Englewood Cliffs, 1966

[11] G. Alefeld, J. Herzberger, "Introduction to interval computations", Academic Press, New York, 1983

[12] E.P. Oppenheimer, A.N. Michel, "Application of Interval Analysis Techniques to Linear Systems: Part I-III", IEEE Trans. on Circuits and Systems, vol. 35, no. 9-10, 1988

[13] E.G. Eszter, R.S. Sanchez Pena, "Value set boundary computation of uncertainty structures", Proc. of the American Control Conference, pp. 2210-13, 1992

[14] Maple V™, Waterloo Maple Software, University of Waterloo, Version 5.2, 1994

[15] R. Tempo, M. Barberis, M. Casales, D. Cavallera, "Robust Stability with Multilinear Pertubations: How Good is the Convex Hull Approximation?", Proc. of the Conf. on Decision and Control, pp. 861-864, Dec 1990